Andreas Franck

Efficient Algorithms for Arbitrary Sample Rate Conversion with Application to Wave Field Synthesis

Efficient Algorithms for Arbitrary Sample Rate Conversion with Application to Wave Field Synthesis

Andreas Franck

Impressum

Only way to feel the noise
is when it's good and loud

Ian Fraser Kilmister

Acknowledgements

This thesis grew during my work at the Fraunhofer Institute for Digital Media Technology IDMT. I would like to thank my superiors and my colleagues at this institute, especially at the Acoustics department, for the inspiring atmosphere, interesting discussions, and their interest in my research work.

I am deeply grateful to my supervisor, Prof. Karlheinz Brandenburg, for his interest and his confidence in this research project, valuable guidance, and insightful discussions.

I would like to thank the reviewers of this thesis, Prof. Julius O. Smith III (CCRMA, Stanford University, California, USA) and Prof. Vesa Välimäki (Aalto University, Finland), for their motivation and encouragement to pursue this work, and for positive feedback and helpful comments on this thesis as well as on earlier papers.

Many people contributed to this work by encouragement and continuing confidence. In the first place, I would like to thank my parents, Bärbel und Johannes Franck. Dr. Klaus Swieczkowski played a major role in convincing me to study an engineering course, to choose Ilmenau, and to continue my research interests. I would also like to thank Dr. Volker Zerbe for his continued motivation to accomplish this thesis.

A very special thank goes to Dr. Alexander Eichhorn, one of my best friends. Many long discussions about our research topics influenced my ways of thinking and approaching this thesis substantially. Furthermore, I would like to thank him for still being skeptical about too many formulas.

Last, but by no means least, I would like to thank my wife Jana and my sons Felix and Johann for love, patience, and for reminding me that there are other and sometimes more important things in life than signal processing.

Abstract

Arbitrary sample rate conversion (ASRC) denotes techniques for changing the sampling rate of discrete-time sequences by arbitrary, potentially time-varying ratios. ASRC is utilized in many applications of digital signal processing.

In this thesis, the application of ASRC methods to wave field synthesis (WFS), a concept for high-quality, spatially correct sound reproduction, is considered. ASRC algorithms may improve the sound quality of moving sound objects significantly. However, the large number of simultaneous ASRC operations in a typical WFS rendering system inhibits the use of sophisticated ASRC algorithms in most cases.

To address this problem, several contributions are proposed in the present work. The computational complexity for WFS is significantly reduced by introducing an algorithm partitioning that enables an efficient reuse of intermediate results. This permits the application of high-quality resampling algorithms with an overall complexity comparable to the use of most basic conventional algorithms. However, this partitioning also poses new requirements to ASRC algorithms and requires trade-offs between different performance measures such as computational complexity, storage requirements, or memory bandwidth.

To improve algorithms and implementation structures for ASRC, several objectives are pursued in this thesis. First, analytical representations for the continuous frequency response of several classes of ASRC structures are introduced. In particular, concise closed-form expressions are derived for Lagrange interpolation, the modified Farrow structure, and structures combining integer-ratio oversampling and continuous-time resampling functions. These representations enable insight into the behavior of the respective structure and are directly applicable to design methods.

A second focus is placed on purposeful coefficient design for these structures, in particular methods that yield optimal results with respect to a selectable error norm and optional design constraints. In contrast to existing approaches, such optimal design methods are proposed also for two-stage ASRC structures consisting of integer oversampling and continuous-time resampling functions. For this class of structures, a set of specifically adapted continuous-time resampling functions is introduced that yield gradual, significant performance improvements when used in conjunction with the proposed overall optimization method.

The variety of implementation structures, each exhibiting several design parameters to be determined, forms a main impediment for the selection of an ASRC algorithm suitable for a given application. For this reason, performance analysis and comparison of different ASRC algorithms forms a third major objective. Analyses are performed to determine the influence of specific design parameters on the achievable quality. Furthermore, the required resources to attain a given design quality are investigated with respect to different performance measures.

In this way, the results of this thesis are not limited to wave field synthesis, but are likely to be applicable to a wide class of applications of arbitrary sample rate conversion.

Zusammenfassung

Verfahren zur unbeschränkten Abtastratenwandlung (arbitrary sample rate conversion, ASRC) ermöglichen die Änderung der Abtastrate zeitdiskreter Signale um beliebige, zeitvariante Verhältnisse. ASRC wird in vielen Anwendungen digitaler Signalverarbeitung eingesetzt.

In dieser Arbeit wird die Verwendung von ASRC-Verfahren in der Wellenfeldsynthese (WFS), einem Verfahren zur hochqualitativen, räumlich korrekten Audio-Wiedergabe, untersucht. Durch ASRC-Algorithmen kann die Wiedergabequalität bewegter Schallquellen in WFS deutlich verbessert werden. Durch die hohe Zahl der in einem WFS-Wiedergabesystem benötigten simultanen ASRC-Operationen ist eine direkte Anwendung hochwertiger Algorithmen jedoch meist nicht möglich.

Zur Lösung dieses Problems werden verschiedene Beiträge vorgestellt. Die Komplexität der WFS-Signalverarbeitung wird durch eine geeignete Partitionierung der ASRC-Algorithmen signifikant reduziert, welche eine effiziente Wiederverwendung von Zwischenergebnissen ermöglicht. Dies erlaubt den Einsatz hochqualitativer Algorithmen zur Abtastratenwandlung mit einer Komplexität, die mit der Anwendung einfacher konventioneller ASRC-Algorithmen vergleichbar ist. Dieses Partitionierungsschema stellt jedoch auch zusätzliche Anforderungen an ASRC-Algorithmen und erfordert Abwägungen zwischen Performance-Maßen wie der algorithmischen Komplexität, Speicherbedarf oder -bandbreite.

Zur Verbesserung von Algorithmen und Implementierungsstrukturen für ASRC werden verschiedene Maßnahmen vorgeschlagen. Zum Einen werden geschlossene, analytische Beschreibungen für den kontinuierlichen Frequenzgang verschiedener Klassen von ASRC-Strukturen eingeführt. Insbesondere für Lagrange-Interpolatoren, die modifizierte Farrow-Struktur sowie Kombinationen aus Überabtastung und zeitkontinuierlichen Resampling-Funktionen werden kompakte Darstellungen hergeleitet, die sowohl Aufschluss über das Verhalten dieser Filter geben als auch eine direkte Verwendung in Design-Methoden ermöglichen.

Einen zweiten Schwerpunkt bildet das Koeffizientendesign für diese Strukturen, insbesondere zum optimalen Entwurf bezüglich einer gewählten Fehlernorm und optionaler Entwurfsbedingungen und -restriktionen. Im Gegensatz zu bisherigen Ansätzen werden solche optimalen Entwurfsmethoden auch für mehrstufige ASRC-Strukturen, welche ganzzahlige Überabtastung mit zeitkontinuierlichen Resampling-Funktionen verbinden, vorgestellt. Für diese Klasse von Strukturen wird eine Reihe angepasster Resampling-Funktionen vorgeschlagen, welche in Verbindung mit den entwickelten optimalen Entwurfsmethoden signifikante Qualitätssteigerungen ermöglichen.

Die Vielzahl von ASRC-Strukturen sowie deren Design-Parameter bildet eine Hauptschwierigkeit bei der Auswahl eines für eine gegebene Anwendung geeigneten Verfahrens. Evaluation und Performance-Vergleiche bilden daher einen dritten Schwerpunkt. Dazu wird zum Einen der Einfluss verschiedener Entwurfsparameter auf die erzielbare Qualität von ASRC-Algorithmen untersucht. Zum Anderen wird der benötigte Aufwand bezüglich verschiedener Performance-Metriken in Abhängigkeit von Design-Qualität dargestellt.

Auf diese Weise sind die Ergebnisse dieser Arbeit nicht auf WFS beschränkt, sondern sind in einer Vielzahl von Anwendungen unbeschränkter Abtastratenwandlung nutzbar.

Contents

List of Symbols

a_{mn} WFS amplitude scaling for virtual source m and secondary source n

A_p Maximum passband variation

A_p Minimum stopband attenuation

B Coefficient matrix of the modified Farrow structure

b_{mn} Coefficient of the modified Farrow structure

$b[n]$ Coefficients of a linear-phase FIR filter

$(b^N)^{-1}[n]$ Inverse discrete B-spline

$B^N(\omega)$ Frequency response of a spline basis function of order N

C Coefficient matrix of the Farrow structure

c Speed of sound

C^k Differentiability class of k times continuously differentiable functions

$C^{[l]}$ Coefficient matrix of the generalized Farrow structure

$C_m(\omega)$ Non-causal frequency response of a subfilter of the modified Farrow structure

$C_m(z)$ Subfilter of the Farrow structure

c_{mn} Coefficients of the Farrow structure

$c_{mn}^{[l]}$ Coefficient of the generalized Farrow structure

$c[n]$ Sequence of spline coefficients

D Fractional delay value

d Fractional part of a fractional delay value

D_{int} Integer part of a fractional delay value

$\text{DTFT}\{x[n]\}$ Discrete-time Fourier transform

$\text{DTFT}^{-1}\{X(\omega)\}$ Inverse discrete-time Fourier transform

$E(\mathrm{e}^{\mathrm{j}\omega}, D)$ Complex error function of a fractional delay element

E_∞ Approximation error with respect to L_∞ norm (minimax or Chebyshev error)

E_2 Integrated squared error

E_p Approximation error with respect to L_p norm

E_{wls} Weighted least squares error

$\mathscr{F}\{\cdot\}$ Continuous Fourier transform operator

$f^{[l]}(m, \mu)$ Polynomial basis function of the generalized Farrow structure

$f(m, \mu)$ Polynomial basis function for the Farrow structure

$\mathscr{F}^{-1}\{\cdot\}$ Inverse continuous Fourier transform operator

$\text{frac}(x)$ Fractional part of x

$g(m, n, \mu)$ Basis function of a symmetric piecewise-polynomial resampling filter, time domain representation

$G(m, n, \omega)$ Basis function of a symmetric piecewise-polynomial resampling filter, frequency domain representation

$G_o(n, \omega)$ Basis function for ASRC systems based on oversampling and a fixed resampling function

$G(\mathbf{x}|\mathbf{x}_s, \omega)$ Green's function for free-field wave propagation

$H_{aa}(j\Omega)$ Continuous-time anti-aliasing filter

$\widehat{H}_c(j\Omega)$ Frequency response of the ideal continuous-time anti-imaging/anti-aliasing filter

$\widehat{h}_c(t)$ Impulse response of the ideal continuous-time anti-imaging/anti-aliasing filter

$H_c(j\Omega)$ Continuous-time frequency response of a combined anti-aliasing/anti-imaging filter

$H_c^{(N)}(j\omega)$ Continuous frequency response of a Lagrange interpolator of order N

$h_c(t)$ Continuous-time impulse response of a combined anti-aliasing/anti-imaging filter

$H_{dig}(e^{j\omega})$ Discrete-time prefilter of ASRC structures based on oversampling

$H\left(e^{j\omega}, D\right)$ Frequency response of a fractional delay element

$\widehat{H}\left(e^{j\omega}, D\right)$ Frequency response of the ideal fractional delay element

$\widetilde{H}^{(k)}(e^{j\omega})$ Ideal discrete-time differentiator of order k

$\widehat{h}(n, D)$ Impulse response of the ideal fractional delay element

$H_{int}(j\omega)$ Continuous-time resampling filter of ASRC structures based on oversampling

$h(k, \mu)$ Filter coefficients of FIR resampling functions

$H(\omega)$ Noncausal frequency response of a linear-phase FIR filter

$H(\omega)$ WFS pre-equalization filter for monopole secondary sources

$H(\omega, \mu)$ Noncausal discrete-time frequency response of the modified Farrow structure

$\widehat{H}_r(j\Omega)$ Frequency response of ideal reconstruction filter for D/C conversion

$\widehat{h}_r(t)$ Impulse response of ideal reconstruction filter for D/C conversion

$H(x)$ Unit step function (or Heaviside step function)

$K(e^{j\omega'}, e^{j\omega})$ Bi-frequency system function

$k[m, n]$ Discrete-time Green's function

$l_k(t)$ Lagrange polynomial

L Integer upsampling ratio in rational or integer-rate sample rate conversion

M Integer downsampling ratio in rational or integer-rate sample rate conversion

mod Modulo operation

$\mathscr{P}_x$ Average signal power of a discrete-time signal

$Q_m(\mathbf{x}, \omega)$ WFS driving function including synthesis operator synthesis operator for a point source and secondary monopole sources

R Sample rate conversion ratio

R^* Multiplication rate for an ASRC system

R^+ Addition rate for an ASRC system

R^{total} Total instruction rate for an ASRC system

round($\cdot$) Round operation, round to the nearest integer

$S(j\Omega)$ Dirac impulse train, frequency domain

$s(t)$ continuous-time Dirac impulse train

$\text{sinc}(x)$ Unnormalized sinc function

T_i Input sampling period of a sample rate converter

$\{t_k\}$ Basepoint set of a resampling filter

T_o Output sampling period of a sample rate converter

$\text{THD}+\text{N}$ Total harmonic noise+distortion

$\text{trig}(m,n,\omega)$ Basis function of the discrete-time frequency response of the subfilters of the modified Farrow structure

$\text{trig}(n,\omega)$ Basis function of the frequency response of linear-phase FIR filters

$W(\omega)$ Weighting function for filter design

$W(\omega,\mu)$ Weighting function for VFD filter design

X Approximation region for ASRC filter design

x_+^n One-sided power function

$X_a(j\Omega)$ Lowpass-filtered continuous-time signal to prevent aliasing in C/D conversion

$x_a(t)$ Lowpass-filtered continuous-time signal to prevent aliasing in C/D conversion

$X_c(j\Omega)$ Continuous-time signal, frequency-domain representation

$x_c(t)$ Continuous-time signal

$X(e^{j\omega})$ Discrete-time signal, frequency domain representation

$x[n]$ Discrete-time sequence

$x^{[n]}$ Central factorial

$x^{\overline{n}}$ Rising factorial

$x(nT_i)$ Discrete-time input sequence in sample rate conversion systems

$x^{\underline{n}}$ Falling factorial

X_p Passband of the approximation region specification

$X_r(j\Omega)$ Reconstructed continuous-time signal, frequency-domain representation

$x_r(t)$ Reconstructed continuous-time signal

X_s Stopband of the approximation region specification

$X_s(j\Omega)$ Pulse amplitude modulated signal, frequency domain

$x_s(t)$ Pulse amplitude modulated continuous-time signal

$y(mT_o)$ Discrete-time output sequence in sample rate conversion systems

$y[n,D]$ Discrete-time output signal of a fractional delay filter

$y(t)$ Continuous-time output signal

$\beta^N(t)$ B-spline basis function of order N

δ^k Central difference operator of order k

δ_p Maximum passband error

δ_s Maximum stopband error

$\delta(t)$ Continuous Dirac impulse

$\eta^N(t)$ Continuous-time impulse response of cardinal spline interpolation of order N

μ Intersample position

μ^* Transformed intersample position for the generalized Farrow structure

ω Normalized angular frequency

Ω_c Cutoff frequency of continuous-time anti-aliasing/anti-imaging filter

ω_c Cutoff frequency

Ω_i Input sampling frequency (angular frequency) of a sample rate converter

Ω_o Output sampling frequency (angular frequency) of a sample rate converter

ω_p Passband edge of a filter design specification

ω_s Stopband edge of a filter design specification

Ω_s Angular sampling frequency

ω_{up} Finite upper frequency limit for ASRC error norms

ϕ Transition bands of the approximation region specification

$\Phi(m,\omega)$ Scaling function of the basis function $G(m,n,\omega)$ of the modified Farrow structure

$\Phi_r(m,\omega)$ Real-valued scaling function of the continuous frequency response of the modified Farrow structure

$\Pi(x)$ Unit rectangle function

$\Psi(m,n,\omega)$ Periodic Part of the basis function $G(m,n,\omega)$ of the modified Farrow structure

$\Psi_r(m,n,\omega)$ Real-valued basis function of the continuous frequency response of the modified Farrow structure

τ Time delay

$\tau_g(\omega)$ Group delay of a discrete-time filter

τ_{mn} WFS delay for virtual source m and secondary source n

$\tau_p(\omega)$ Phase delay of a discrete-time filter

$\Theta(\omega)$ Phase response of a discrete-time filter

$*$ Convolution operator

∇^k Backward difference operator of order k

Chapter 1

Introduction

This thesis deals with two subjects — arbitrary sample rate conversion and wave field synthesis. While the former is a general problem of digital signal processing, the latter is a spatial sound reproduction concept. Why should it make sense to handle these distinct problems, one of pure and one of applied research, in a single treatise? In the author's opinion, both subjects benefit from this combination.

Wave field synthesis (WFS) is a sound reproduction concept that aims at synthesizing physically correct wave fields of complex auditory scenes consisting of multiple sound sources using a typically large number of loudspeakers. This enables superior localization properties over an extended listening area. Among other advantages, the ability to reproduce moving sound sources realistically forms a distinguishing feature of WFS. Practical sound reproduction is performed by applying so-called WFS operators and is generally implemented as a discrete-time signal processing system. At the signal processing level, the WFS operators essentially calculate delay and scale parameters for each combination of sound sources and loudspeakers and apply these parameters to discrete-time source signals. Due to the large number of simultaneous operations, WFS rendering is a computationally complex task. In case of moving sound sources, the scale and delay parameters are time-varying, resulting in a further increase of the computational requirements.

Arbitrary sample rate conversion (ASRC) refers to converting a discrete-time sequence to another sequence with a different sampling rate. Opposed to other approaches to sample rate conversion, ASRC enables arbitrary conversion ratios. Furthermore, most methods allow this ratio to be changed dynamically. To do so, ASRC algorithms require the capability to approximate the value of a signal represented by a discrete-time sequence at arbitrary points in time, that is, at any position between input samples. Numerous algorithms and implementation structures have been proposed for ASRC which differ in quality and the computational cost required.

So, why does it make sense to consider these distinct topics in combination? The most apparent reason is that WFS is an application of ASRC. As will be shown later, the time-varying delay operation in a WFS system is most appropriately modeled by an ASRC process. It is important to note, however, that this interpretation applies only to the way the time-varying delays are modeled internally. From the external point of view, WFS signal processing forms a single-rate discrete-time system that does not perform an explicit change of the audio sampling rate.

In any case, WFS is a very demanding application of ASRC techniques for two primary reasons. First, it requires the simultaneous application of different, continuously time-varying conversion ratios, covering both sample rate increases and decreases.

Second and more important, the large number of simultaneous ASRC operations in a typical

WFS rendering system imposes very strict requirements on the computational complexity of the algorithms. To reach this objective, several measures are taken in this work. First, strong emphasis is placed on efficient ASRC algorithms and implementation structures. Design methods for these structures that yield optimal filter coefficients with respect to a given design specification form a second objective.

Notwithstanding these improvements in efficiency, the large number of simultaneous ASRC operations in a WFS rendering system inhibits the application of conventional algorithms in most cases. For this reason, a processing structure is proposed that integrates several widely-used classes of ASRC algorithms into the signal flow of a WFS rendering system, resulting in a significant reduction of the computational effort. Obviously, such complexity reductions are possible only if WFS and ASRC techniques are considered in combination.

Finally, the application to a demanding, practical signal processing problem emphasizes performance aspects that are relevant to actual, real-world hardware platforms. While this thesis does not presume a particular target architecture, a set of abstract performance measures is proposed to assess the appropriateness of ASRC algorithms for different hardware platforms.

However, these improvements are not limited to WFS, but are likely to be applicable to a wide range of signal processing problems incorporating sample rate conversion. The utility of efficient implementation structures and methods for optimal coefficient design is self-evident. Furthermore, extensive performance investigations and trade-off analyses enable the assessment of various ASRC algorithms with respect to different performance metrics. In this way, a main intention of this work is to aid an engineer in selecting an ASRC algorithm that is optimally suited for a given target architecture and meets the requirements of an application with minimal effort.

1.1 Outline and Contributions

The remainder of this thesis is structured as follows:

Chapter 2 provides a brief characterization of WFS. The signal processing tasks in a WFS signal processing system are described, emphasizing the central role of time-varying delay operations. Moreover, features and implications of moving sound sources in WFS are characterized.

In Chapter 3, existing approaches to arbitrary-valued delay operations on discrete-time signals are reviewed. Starting from the conversion between continuous-time and discrete-time signals, two approaches to arbitrary delay operations are characterized: Variable fractional delay (VFD) filtering and arbitrary sample rate conversion (ASRC). For these approaches, the most important design methods and implementation structures are reviewed. Particular emphasis is put on a uniform representation for both approaches, thus allowing implementation structures and design methods to be described consistently. Moreover, the computational efficiency of the different implementation structures forms a main focus.

The application of time-varying delay algorithms to WFS is considered in Chapter 4. It is shown why ASRC is more appropriate to model this delay operation. Signal processing structures are proposed which enable the use of high-quality ASRC algorithms while mitigating the growth of the computational complexity. Baseband aliasing artifacts are an additional issue in the application of time-variant delays to WFS. Section 4.4 discusses conditions that require an explicit handling of this problem, and proposes a resource-efficient solution.

Several improvements to ASRC algorithms and implementation structures are developed and analyzed in Chapter 5. First, closed-form expressions for the continuous frequency response of two widely-used structures, namely the modified Farrow structure and the Lagrange interpolator, are derived for arbitrary orders. Based on these derivations, a flexible framework for coefficient design subject to a selectable error norm and flexible optional design specifications is presented. Two-stage structures consisting of integer-ratio oversampling and continuous-time resampling functions are considered in Sections 5.4 to 5.6. Starting from the widely-used structure based on oversampling and Lagrange interpolation, it is shown that the performance can be improved by design methods that take the frequency response of the overall structure into account. At the same time, this optimization scheme enables optimal coefficient design with respect to a specified error norm and optional constraints. Further improvements to this structure are achieved by replacing Lagrange interpolation by more appropriate resampling filters. In Section 5.6, design techniques for the joint optimization of the oversampling component and the continuous-time resampling filter are investigated.

Chapter 6 is concerned with performance analysis and comparisons of the implementation structures considered in this thesis. In the first part, the performance of the different structures is analyzed as a function of several design parameters and specifications. In the second part, trade-off analyses are performed with respect to different performance metrics, thus enabling a purposeful selection of ASRC algorithms appropriate for a given application.

Chapter 7 concludes this thesis by summarizing the main results and points out possible directions of future research.

Chapter 2

Motivation: Wave Field Synthesis

Wave field synthesis (WFS) is a sound reproduction concept that enables excellent localization over an extended listening area. Unlike many other spatial sound reproduction techniques, it aims at a physically correct synthesis of a desired wave field. Typically, WFS is based on a large number of loudspeakers, which are referred to as secondary sources. They are organized in planar or linear arrays enclosing the listening area. WFS is generally implemented using discrete-time signal processing. It is capable of reproducing complex auditory scenes consisting of numerous acoustic objects, which are typically denoted as primary or virtual sources.

The concept of WFS has been initially proposed at the Delft University of Technology in the late 1980s and in the 1990s, starting with several publications, e.g. [Ber88, BdVV93], and further investigated in a series of dissertations, including [Vog93, Sta97, Ver97, Son00, Hul04]. Since then, WFS has developed into an area of active research, conducted at numerous research institutions and in several research projects, e.g. [SPB01]. Nowadays, a large number of WFS installations are in use, from research setups to commercial applications in theme parks, cinemas or open-air live reproduction systems [BBS04a, dV09].

2.1 Theory of WFS

The present thesis does not attempt to provide an in-depth review of the theoretical foundations of WFS. Instead, only the general ideas and the concepts relevant to WFS signal processing are introduced. For more information, the reader is referred to the original publications, the dissertations mentioned above, and the recent overviews [SRA08, dV09].

Conceptually, WFS is based upon the Kirchhoff-Helmholtz integral equation

$$P(\mathbf{x}, \omega) = \frac{1}{4\pi} \int_S \frac{\partial}{\partial \mathbf{n}} G(\mathbf{x}|\mathbf{x}_s, \omega) P(\mathbf{x}_s, \omega) - G(\mathbf{x}|\mathbf{x}_s, \omega) \frac{\partial}{\partial \mathbf{n}} P(\mathbf{x}_s, \omega) \, dS \,. \tag{2.1}$$

This representation theorem states that the acoustic pressure $P(\mathbf{x}, \omega)$ for an angular frequency ω at any point $\mathbf{x}$ within a source-free volume V is determined by the acoustic pressure and the particle velocity at the surface S enclosing V. The pressure at a point $\mathbf{x}_s$ on the surface S is denoted by $P(\mathbf{x}_s, \omega)$. The directional derivative $\frac{\partial}{\partial \mathbf{n}} P(\mathbf{x}_s, \omega)$ is proportional to the normal component of the particle velocity with respect to the inward pointing surface normal $\mathbf{n}$ at the point $\mathbf{x}_s$. The underlying geometry is shown in Figure 2.1. In this representation, $\Psi(\mathbf{x}, \omega)$ represents a distribution of acoustic sources outside of V.

The term $G(\mathbf{x}|\mathbf{x}_s, \omega)$ denotes a Green's function, that is, an elementary solution of the inhomogeneous wave equation governing the propagation of sound. It can be considered as the
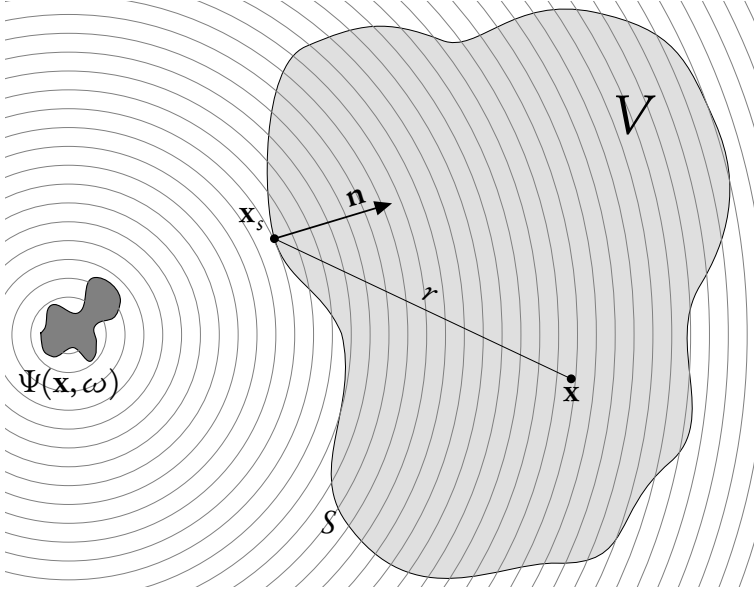
Figure 2.1: Geometry underlying the Kirchhoff-Helmholtz integral theorem. $\Psi(\mathbf{x}, \omega)$ denotes a source distribution outside the source-free volume V.

pressure field of an ideal monopole source driven by a harmonic signal with a single frequency ω. In this way, the Green's function represents the transfer function of a monopole source located at $\mathbf{x}_s$ to the receiver position $\mathbf{x}$. In case of WFS, free-field propagation is generally assumed. While there are several degrees of freedom in the choice of a Green's function, the most simple frequency-domain representation is given by

$$G(\mathbf{x}|\mathbf{x}_s, \omega) = \frac{e^{-j\frac{\omega}{c}r}}{r} \qquad \text{with } r = |\mathbf{x} - \mathbf{x}_s|, \tag{2.2}$$

where r is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_s$. c denotes the speed of sound. In an identical fashion, the transfer function of an elementary dipole source is expressed using the directional derivative $\frac{\partial}{\partial \mathbf{n}} G(\mathbf{x}|\mathbf{x}_s, \omega)$.

The Kirchhoff-Helmholtz integral can also be used as a conceptual basis for sound reproduction. For this purpose, continuous distributions of monopole and dipole sources, which are referred to as *secondary sources*, are placed on the surface S enclosing V. The dipoles are aligned to the inward pointing normal vector $\mathbf{n}$ of the surface. They are driven by the acoustic pressure $P(\mathbf{x}_s, \omega)$ of the wave field of a primary source distribution $\Psi(\mathbf{x}, \omega)$. Likewise, the elementary monopoles are controlled by the directional derivate $\frac{\partial}{\partial \mathbf{n}} P(\mathbf{x}_s, \omega)$ of the sound pressure, corresponding to the normal component of the particle velocity. In this way, the secondary sources reproduce the pressure field of a primary source distribution $\Psi(\mathbf{x}, \omega)$ inside the source-free volume V. This is the basic idea underlying wave field synthesis.

For practical sound reproduction, a series of simplifications is applied, which require additional conditions to hold and generally introduce reproduction errors. The main simplification steps are:

Restriction to monopole or dipole sources   By imposing free-field conditions and restricting the surface S to a plane of infinite extent, the Kirchhoff-Helmholtz integral can be stated as a Rayleigh integral of either type I (2.3a) or II (2.3b). These representations require distributions of only monopole or dipole sources, respectively

$$P(\mathbf{x}, \omega) = \frac{1}{2\pi} \int\limits_{S} -\frac{\partial}{\partial \mathbf{n}} P(\mathbf{x}_s, \omega) \frac{\mathrm{e}^{-\mathrm{j}\frac{\omega}{c} r}}{r} \, \mathrm{d}S \tag{2.3a}$$

$$P(\mathbf{x}, \omega) = \frac{1}{2\pi} \int\limits_{S} P(\mathbf{x}_s, \omega) \frac{1 + \mathrm{j}\frac{\omega}{c} r}{r} \cos(\phi) \frac{\mathrm{e}^{-\mathrm{j}\frac{\omega}{c} r}}{r} \, \mathrm{d}S . \tag{2.3b}$$

In (2.3b), ϕ denotes the angle between $\mathbf{n}$ and the vector $\mathbf{x} - \mathbf{x}_s$. In addition to the restrictions noted above, the surface S emanates a mirrored wave field on the outside of the volume V. In contrast, synthesis based on the Kirchhoff-Helmholtz integral results in a cancellation of the radiation of the secondary monopole and dipole sources outside V.

Restriction to linear secondary source distributions   For most applications, planar 2-dimensional distributions of secondary sources are unfeasible due to architectural or economical considerations. Therefore, they are typically reduced to a horizontal array of either monopole or dipole sources, which are driven by so-called $2\frac{1}{2}$-D synthesis operators [Sta97, Ver97]. For this purpose, the infinite plane S is partitioned into vertical slices of infinitesimal width. Subsequently, the sound field of each slice is approximated by a single elementary source located in the horizontal plane of the secondary source distribution. This derivation makes use of the so-called *stationary phase approximation* (e.g. [Ble84]), a tool for the approximate integration of certain types of oscillatory integrals of the form (2.3).

The transformation into a linear distribution of secondary sources gives rise to several restrictions and errors. First, correct reproduction is restricted to sound sources and listeners both situated in the horizontal plane of the loudspeaker array. Second, the approximation is sufficiently exact only if the distances of both the source and the receiver point to the secondary source distribution are large compared to the wavelength $\lambda = \frac{2\pi c}{\omega}$ [SA10]. Finally, the amplitude of the synthesized wave field is correct only at the so-called *receiver line*, a line or curve in the reproduction plane that can be chosen freely within certain limits.

Discretization of the secondary source distribution   For practical sound reproduction, the continuous distribution of secondary sources is replaced by arrays of discrete loudspeakers with finite spacing. As a consequence, the wave field cannot be reproduced correctly above a given signal frequency that is determined by the distance between adjacent secondary sources. This effect is closely related to aliasing artifacts caused by a discrete sampling of continuous-time signals. For this reason, this artifact is termed *spatial aliasing*. Above the spatial aliasing frequency, magnitude deviations and sound colorations occur that may fluctuate rapidly as a function of both the source and the listener position.

Finite secondary source distributions   Finally, the secondary source distributions, which were assumed infinite so far, are truncated to a finite extent in realizable setups. This may cause

amplitude errors and additional wave fronts emanating from the end points of the array. The latter artifacts are often termed *diffraction effects* [Sta97] due to their resemblance with the diffraction of acoustic waves at edges.

Additional errors of realizable secondary source distributions   In practical applications, several additional sources of errors are introduced. For instance, the linear arrays are typically replaced by loudspeaker configurations enclosing the listening area, which often also contain gaps. Moreover, the characteristics of the secondary speakers typically deviate from the assumed ideal monopole or dipole characteristics.

For most of the problems described in this section, techniques have been proposed to reduce the audible artifacts. However, these improvements are often applicable only to a limited set of loudspeaker geometries or source positions.

2.2 WFS Synthesis Operators

The driving signals for the discrete secondary sources are obtained by applying so-called *synthesis operators* to the signal of the acoustic object (the *virtual sound source* or *primary source*) to be reproduced. The synthesis operator is specific to the type of the virtual source and incorporates properties such as the source position.

In typical WFS reproduction systems, synthesis operators are implemented for elementary source characteristics such as monopoles, i.e., point sources, or plane waves. Extensions have been proposed to generalize the synthesis operator to arbitrary directivity characteristics [Cor07, AS07], but these developments are yet on a theoretical level that does not allow an immediate application to arbitrary source characteristics or arbitrary time-domain signals.

As an example, the driving function of a monopole source outside the listening area for an arbitrary receiver contour R is given by

$$Q_m(\mathbf{x}_s, \omega) = \underbrace{S(\omega)}_{\text{Source signal}} \underbrace{\sqrt{\frac{\mathrm{j}\omega}{2\pi c}} \sqrt{\frac{\Delta r}{r + \Delta r}} \frac{1}{\sqrt{r}} \cos\phi \, \mathrm{e}^{-\mathrm{j}\frac{\omega}{c} r}}_{\text{Synthesis operator}} \tag{2.4}$$

$$\text{with} \quad r = |\mathbf{x}_\Psi - \mathbf{x}_s|, \ \Delta r = |\mathbf{x}_s - \mathbf{x}_R|.$$

The corresponding geometry is shown in Figure 2.2. The receiver point $\mathbf{x}_R$ is formed by the intersection of the line through the source position $\mathbf{x}_\Psi$ and the position of the secondary source $\mathbf{x}_s$ with the receiver contour R. ϕ is the angle between this line and the normal vector $\mathbf{n}$ of the secondary source at $\mathbf{x}_s$.

The driving function (2.4) consists of several components that represent distinct functionalities. While $S(\omega)$ constitutes the frequency-domain characteristics of the source signal, the term

$$H(\omega) = \sqrt{\frac{\mathrm{j}\omega}{2\pi c}} \tag{2.5}$$

denotes a filtering operation that is independent of the position of the virtual source. It is referred to as WFS pre-equalization filter in [SA10]. For monopole secondary sources, it represents a
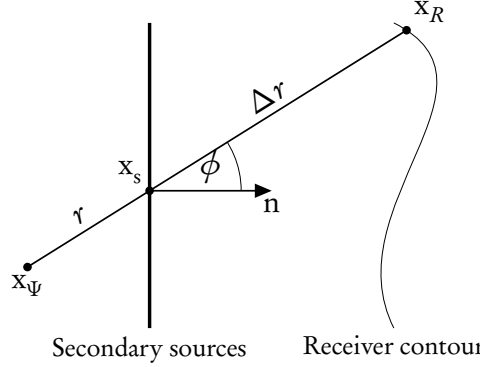
Figure 2.2: Geometry for the WFS synthesis operators with arbitrary receiver contour R.

lowpass filter with a constant slope of 3 dB/octave, while it forms a highpass with a magnitude increase of 3 dB/octave in case of dipole secondary sources. The term

$$a = \sqrt{\frac{\Delta r}{r + \Delta r}} \frac{1}{\sqrt{r}} \cos \phi \qquad (2.6)$$

denotes an amplitude factor that depends on the positions of the primary and secondary sources and the receiver point. Finally, $e^{-j\frac{\omega}{c}r}$ represents a time delay

$$\tau = \frac{|\mathbf{x}_\Psi - \mathbf{x}_s|}{c}, \qquad (2.7)$$

which is proportional to the distance between the virtual and the secondary source.

2.3 Signal Processing for WFS

In WFS reproduction systems, the transformations represented by the synthesis operators are generally implemented by means of discrete-time signal processing. Mainly due to the typically large number of loudspeakers and the benefits of complex auditory scenes consisting of multiple virtual sources, signal processing for WFS is a computationally expensive task.

However, research on WFS is predominantly focused on theoretical models, such as the synthesis operators for single virtual sources and their influence on the properties of the synthesized wave field. The corresponding signal processing is handled very scarcely in literature. Brief characterizations are found, for instance, in [Sta97, Ver97, Jan97].

Real-time WFS reproduction systems typically operate in the time domain. An inverse Fourier transform of the synthesis operator (2.4) yields the time-domain signal component $y_n^{(m)}(t)$ for a secondary source n obtained from the source signal $x_m(t)$ of a single virtual source denoted by the index m

$$y_n^{(m)}(t) = a_{mn} \cdot x_m(t) * h(t) * \delta \left(t - \tau_{mn} \right). \qquad (2.8)$$

Here, $*$ denotes the continuous convolution operator. In the following, $*$ will be used for both continuous and discrete convolution, because the distinction is generally clear from the context. In (2.8), $h(t)$ is the continuous-time impulse response of the pre-equalization filter $H(\omega)$ (2.5), and $\delta(t - \tau_{mn})$ represents a continuous-time delay by the delay value τ_{mn}. The coefficients a_{mn} and τ_{mn} represent the amplitude and delay coefficients as defined in (2.6) and (2.7), respectively, for the combination of a virtual source m and a secondary speaker n. These coefficients are referred to as *WFS coefficients* in the following.

The loudspeaker driving signal is formed by superimposing the signal components $y_n^{(m)}(t)$ of all virtual sources

$$y_n(t) = \sum_{m=0}^{M} y_n^{(m)}(t). \tag{2.9}$$

In a discrete-time signal processing system with sampling frequency f_s, the signals as well as the convolution operators are replaced by their discrete-time counterparts, resulting in

$$y_n[k] = \sum_{m=0} a_{mn} \cdot x_m[k] * h[k] * \delta[k - \tau_{mn}f_s]. \tag{2.10}$$

The discrete-time delay operation $\delta[k - \tau_{mn}f_s]$ is used here in a symbolic way. Implications and actual implementations of this delay operator will be investigated later.

Consequently, the signal processing algorithm for WFS consists of several distinct operations:

- The calculation of the scale and delay coefficients a_{mn} and τ_{mn} is defined by the synthesis operator (2.4). Because the properties of a virtual source such as the position $\mathbf{x}_s$ may vary over time, these calculations must be repeated periodically or on demand. However, these properties of a virtual source change relatively slowly in general. Therefore, it is sufficient to calculate the WFS coefficients with update rates much lower than the sampling frequency of the system [Jan97, Ver97]. In general, the application of simple interpolation techniques is sufficient to ensure smooth trajectories between parameter updates.

- The WFS pre-equalization filter $H(\omega)$ (2.5) is generally implemented as a discrete-time IIR or FIR filter. In a conventional WFS system, the filter response is independent of the secondary speaker as well as of the properties of the sound source. Therefore, it is sufficient to perform this operation only once for each input signal $x_m[k]$. Alternatively, this equalization can be applied to the accumulated output signals, which proves sensible in some implementations. For instance, this filter characteristic might be integrated into a loudspeaker-specific equalization filter.

- The delay term $\delta[k - \tau_{mn}f_s]$ is applied to each combination of primary and secondary sources. In a discrete-time system, this operation is conveniently modeled by a variable-length delay line (e.g. [Roc00, Zöl02, Smi10a]) for each source signal $x_m[k]$. This delay line must enable multiple accesses for the computation of all signal components $y_n^{(m)}[k]$. The required storage capacity of these delay lines is mainly determined by the maximally allowed distance between a primary and a secondary source.

- Similar to the delay operator, the scaling coefficient a_{mn} is applied to each combination of primary and secondary sources. For this reason, these operations are combined in most implementations. This combination will be denoted as *scale&delay* in the following.
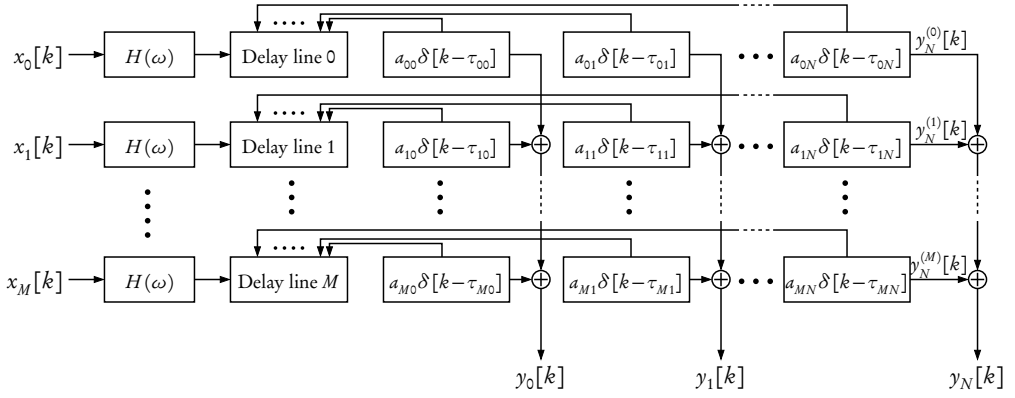
Figure 2.3: WFS signal processing system (adapted from [Sta97]).

- The scale&delay operation must be performed at the audio sampling frequency of the system. Moreover, the number of required simultaneous operations is given by the product of the numbers of primary and secondary sources. For these reasons, the scale&delay operation is typically the most critical part in a real-time WFS reproduction system. For instance, the effort for this operation accounts for more than 60 % of the complete computational load in an efficient WFS rendering system, even if no interpolation of WFS coefficients is performed and all delays are implemented as integer-valued index shifts.
  Techniques to reduce the number of scale&delay operations by excluding component signals that do not contribute to the desired wave field have been proposed, for instance, in [Ver97]. However, the number of active loudspeakers for a given primary source is typically reduced only to a constant fraction. Thus, the asymptotic number of operations $O(MN)$ for a system with M primary and N secondary sources remains unchanged. Moreover, a constant computational complexity with good worst-case behavior is more important for real-time rendering systems than good performance in the best or average case.
- The accumulation of the signal components for each loudspeaker signal combines the results of the scale&delay operations. It is typically tightly integrated into this part of the algorithm, for instance by the use of multiply-and-accumulate functionalities provided by most DSP or general-purpose hardware architectures. For this reason, it is sensible not to consider this operation separately, but to regard it as a part of the scale&delay operation instead.

Figure 2.3 depicts the general signal flow of the audio signal processing component in a WFS reproduction system. This scheme is based on [Sta97], but the variable delay lines and the arbitrary delay line accesses by the scale&delay operators are shown explicitly.

2.4 Moving Sound Sources

The ability to reproduce moving sound sources is beneficial for many applications of spatial sound reproduction, for instance in movie theatres, theme parks, live performances, and immersive

virtual-reality applications [Boo01a, Spo04, BBS04b].

From the conceptual point of view, WFS is well suited for the synthesis of moving sound objects. On the one hand, the abilities to synthesize virtual sound sources at arbitrary positions and to reproduce complex auditory scenes form a sound technological basis. On the other hand, the excellent localization properties over an extensive listening area without requiring headphones or tracking techniques makes WFS an advantageous reproduction technology for moving sources, especially for larger audiences.

2.4.1 The Doppler Effect

The Doppler shift or Doppler effect is probably the most characteristic property of a moving sound source. Depending on the speed of the sound source and/or the listener with respect to the medium, the perceived frequency of a sound signal is altered [Str00, Zöl02, SSAB02, Smi10a]. Intuitively speaking, this effect is caused by the contraction or dilation of the acoustic wave fronts due to the varying traveling time of the sound. For a monochromatic sound source with signal frequency f_{src}, the perceived frequency at the listener position is given by

$$f_{rec} = \frac{c + v_{rec}}{c - v_{src}} f_{src}, \tag{2.11}$$

where v_{src} and v_{rec} denote the scalar velocities of the source and the receiver with respect to the propagation medium.

In a WFS reproduction system, the movement of a virtual source causes an effect that resembles the natural Doppler effect. As a result of the movement, the distances between the primary source and the secondary sources change, resulting in dynamically changing delay term τ_{mn} (2.7) of the driving function (2.4). Conceptually, these changing delays are equivalent to the varying traveling times causing the natural Doppler effect. In this way, the wave fields emitted by the secondary sources exhibit a frequency shift, and the effect perceived at the listener position resembles the sound of the moving primary source.

It is noted that, while physically correct, the Doppler effect is not necessarily desirable in sound reproduction systems. For instance, while acoustic scenes consisting of moving instruments or voices can be used advantageously in many applications, the effect of the accompanying time-variant Doppler shift is often distracting, especially for tonal content. However, for sound reproduction concepts such as WFS that use time delays systematically, the effects caused by changes of these delays cannot be suppressed trivially. In any case, a correct synthesis of the Doppler effect appears to be a sensible starting point to control and alter this effect of WFS rendering purposefully.

2.4.2 Artifacts of Moving Sound Sources

Despite the conceptual benefits, the synthesis of moving virtual sources causes several distinct audio artifacts in realizable WFS reproduction systems. Analyses of these errors have been performed in [FGKS07, AS08]. Therefore, they are characterized here in brief form only.

First, the artifacts associated with the synthesis of static, non-moving sound sources described in Section 2.1, such as spatial aliasing, amplitude errors, or diffraction effects, apply to moving sound objects as well. Moreover, as most of these artifacts depend on the position of the virtual
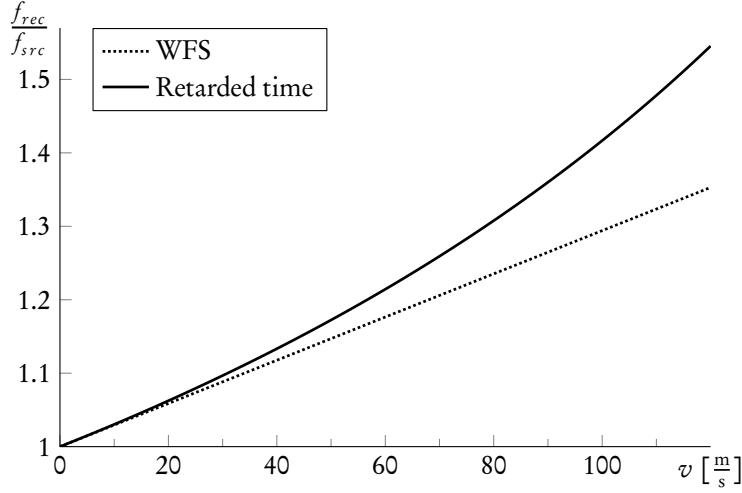
Figure 2.4: Doppler shift in the driving function of a secondary source for a directly approaching primary source.

source, these errors fluctuate over time for moving sound sources. In most cases, such time-variant artifacts are more noticeable or disturbing than steady, constant deviations. For instance, the amplitude errors and colorations caused by spatial aliasing fluctuate rapidly for moving sound objects.

In addition to these errors originating in the synthesis of static sound sources, several artifacts are specific to moving sound sources.

Systematic deviation of the Doppler shift   In the conventional time-domain driving function (2.8), the calculation of the time delay τ_{mn} (2.7) is based on the current position of the virtual sound source $\mathbf{x}_\Psi$. However, for natural moving sources, the travel time of the sound event as well as the position of emission differ from these values due to the movement during this travel time. The actual travel time is commonly referred to as *retarded time* $\tau(t)$ [Str98, Str00], a notion widely used in the field of electrodynamics [Jac02]. In the general case, the retarded time is determined by an algebraic equation involving the time-variant source position $\mathbf{x}_\Psi(t)$

$$c\tau(t) = |\mathbf{x}_s - \mathbf{x}_\Psi(t - \tau(t))|. \tag{2.12}$$

Analytic solutions to this equation exist only for specific source trajectories such as linear paths. However, it is assumed that for sufficiently regular source movements, which can be approximated for instance by piecewise linear or polynomial segments, the retarded time can be determined to an acceptable accuracy with moderate computational effort.

As the retarded time depends on the coordinates of the moving source, neglecting this effect alters the Doppler shifts synthesized by the secondary sound sources. However, for moderate velocities, this error is relatively small. It is notable that the calculation of the traveling time

(a) Geometry   (b) Doppler shifts
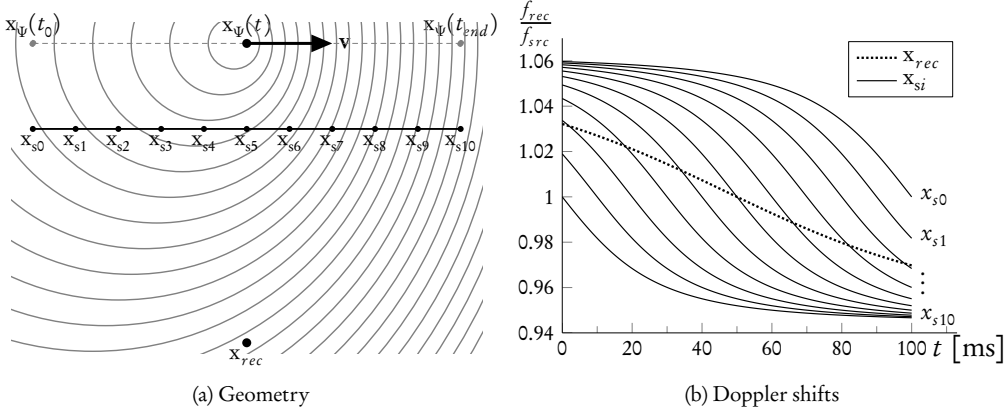
Figure 2.5: The Doppler shifts of a natural moving sound source and of the secondary source signals for a uniform movement parallel to a linear array. Source velocity $|\mathbf{v}| = 20\,\text{m/s}$, secondary speaker distance $\Delta x = 0.2\,\text{m}$.

based on the current source position corresponds to a Doppler shift

$$f_{rec} = \frac{c + v}{c} f_{src},$$

which is a special case of (2.11) for a receiver moving with velocity v while the source is static with respect to the propagation medium [SSAB02].

In Figure 2.4, the ratio between the emitted and the received frequency is shown as a function of the velocity v for a virtual source directly approaching a secondary source. This movement constitutes the worst-case deviation from the retarded time for a given source velocity.

It is observed that the difference between the conventional driving signal and an adopted synthesis operator that incorporates the retarded time is relatively small. For this reason, it is questionable whether this artifact is important in practical WFS systems. In particular, in most scenarios that make purposeful use of moving sound sources, the qualitative impression of the Doppler effect is more important than a quantitatively correct frequency shift. On the other hand, because the retarded time affects the driving signals of the secondary sources, it is important to investigate and assess its influence on the reproduced sound field, which is formed by the superposition of a large number of secondary sources.

Spectral Artifacts   In actual reproduction scenarios as well as in numerical simulations, an effect has been observed that manifests as a broadening or spread of the Doppler-shifted spectrum of the primary sound source. This artifact is often perceived as a beat [Grä07, FGKS07]. The magnitude of the effect increases with the velocity of the virtual sound source and decreases with increasing distance between the primary and secondary sources. The perceived frequencies coincide approximately with the frequency-shifted individual driving signals of the secondary sources.

The distinct Doppler shifts of the secondary sources are illustrated in Figure 2.5 for a source movement parallel to a linear loudspeaker array.

It has been observed in [FGKS07] that the spectrum of the synthesized wave field resembles the combination of the distinct Doppler shifts of the secondary sources. Therefore, it has been conjectured that the superposition of the secondary wave fields differs from the wave field of the primary source due to restrictions and simplifications introduced in the derivation of the WFS synthesis operators.

However, this effect has not been investigated analytically so far. In [AS08], an alternative interpretation of the perceived artifacts has been stated, attributing these effects to diffraction artifacts due to the finite extent of the secondary source array.

In any case, a precise quantitative analysis of this effect, both in acoustic measurements and in simulations, bears several practical as well as theoretical difficulties. For instance, artifacts of static WFS reproduction, such as diffraction effects, influences of the reproduction room, or spatial aliasing, are also present and influence the obtained data. Second, to create a pronounced effect, the source velocity is required to be relatively high and to last for a sufficiently long time. Therefore, the array of secondary sources must be very long in order to get a meaningful wave field over this period of time without introducing diffraction artifacts.

Finally, as the process is time-variant, the resulting spectrum has to be examined using time-frequency analysis techniques [Coh95, Smi10b]. Due to the uncertainty principle of time-frequency analysis, a trade-off between time and frequency resolution has to be found. Furthermore, the numerous parameters involved in the time-frequency analysis, for instance the window function or the amount of overlap, influence the correctness and the expressiveness of the results.

Signal Processing Artifacts For practical WFS reproduction systems, the effects of the time-variant properties of moving sound sources on the discrete-time implementation must be taken into account. Here, the discrete-time approximation of the delay operator $\delta[k - \tau_{mn} f_s]$ in (2.10) turns out to be of paramount importance. A direct discrete-time implementation based on rounding the delay to the nearest integer multiple of the sampling period results in severely audible artifacts, which are described as "sputtering" in [Ver97, Jan97]. Thus, sophisticated discrete-time algorithms are required to synthesize arbitrary delay values.

In the context of general signal processing and audio applications, the time-variant delay operation is equivalent to the application of variable-length delay lines. Writes and/or reads to such structures require sophisticated interpolation techniques for acceptable audio quality [Dat97, Roc00, Zöl02, Smi10a].

2.5 Objective: High-Quality Algorithms for Variable Delays for WFS

In this chapter, it became apparent that the delay operator as part of the scale&delay operation is most performance-critical for WFS rendering systems due to the large number of simultaneous operations. Thus, the cost of high-quality algorithms to synthesize discrete-time sequences with time-varying delays is often prohibitive.

Therefore, in order to develop efficient signal processing algorithms for WFS, the main objectives pursued in this thesis are as follows: First, an appropriate model for the delay operation in WFS signal processing has to be determined. Second, the selection of suitable algorithms and implementation structures and the corresponding design techniques are of prime importance. Third, improvements to these algorithms are investigated, as they directly influence the quality and the computational complexity of WFS reproduction systems. Finally, it is necessary to integrate these algorithms into the signal processing scheme for WFS in an efficient way.

It should be emphasized that the necessity for applying arbitrary delay values is principally induced by the reproduction of moving sound sources. For static sound sources, rounding or truncating the delay value to the next integer multiple of the sampling period causes a slight deviation in the synthesis operator (2.4). The effect of this difference is similar to small displacements of the secondary speakers with respect to the actual coordinates $\mathbf{x}_s$ used in the synthesis operator. For instance, the maximum delay error in a reproduction system with sampling frequency $f_s = 48\,\mathrm{KHz}$ caused by rounding to the nearest unit sample delay corresponds to a displacement of about 3.5 mm in the normal direction of a secondary source. According to [SM07], such small variations are unlikely to produce noticeable effects. An analysis of WFS reproduction with static sound sources and fractional-sample delay algorithms [AGS10], which also includes subjective listening tests, supports this assessment.

The artifacts due to signal processing are only one of several inaccuracies in the reproduction of moving sound sources using WFS. Therefore, the algorithms developed here solve only one problem associated with the reproduction of dynamic auditory scenes. Moreover, it is possible that the advances are partly masked by other artifacts. However, most of the other errors are of conceptual nature, or their solution would require a disproportionate effort such as planar, dense loudspeaker distributions. In contrast, the errors related to signal processing can be alleviated by algorithmic means and thus in a cost-efficient manner.

Chapter 3

State of the Art: Delaying Discrete-Time Sequences

3.1 Introduction

As pointed out in the preceding chapter, delaying discrete-time signals by arbitrary, time-varying delay values is one of the central signal processing operations in a WFS reproduction system. This operation is an area of extensive research in the field of digital signal processing. It becomes apparent that there are two essential approaches to this delay operation: Fractional delay (FD) filtering and arbitrary sample rate conversion (ASRC). In this chapter, the research in these areas is reviewed.

Conceptually, applying an arbitrary-valued delay on a discrete-time signal is appropriately modeled as the reconstruction of a continuous-time signal from a bandlimited discrete-time sequence followed by a sampling process. Consequently, this operation is often termed *bandlimited interpolation* [Smi11, Smi10a]. Therefore, the basic principles of sampling and reconstructing continuous-time signals are reviewed in Section 3.2. However, it is important to note that, although the delay operation conceptually involves continuous-time signals, it is generally implemented as a purely discrete-time system.

In Section 3.3, the fractional delay filtering approach to arbitrary delays is reviewed. To this end, its definition, applications, error measures and properties are summarized. In the same style, arbitrary sampling rate conversion is characterized in Section 3.4.

Despite the conceptual differences between these approaches, both are generally implemented as discrete-time filtering processes, which are described in Section 3.5. Here, a common notation, namely the use of basepoint sets and the intersample position, is emphasized. On the one hand, this enables a uniform characterization of numerous algorithms. On the other hand, this notation clarifies several general properties of ASRC and FD algorithms.

Based on this description, algorithms and implementation structures are reviewed in Section 3.6. It becomes apparent that, notwithstanding the differences between FD and ASRC, the implementation structures are identical in most cases. In contrast, coefficient design for these structures is different for both approaches. Therefore, design methods are reviewed in Sections 3.7 and 3.8, respectively.

The FD and ASRC approaches to arbitrary delay operations on discrete-time signals are compared in Section 3.9, both from the analytical viewpoint and from practical design and application considerations.
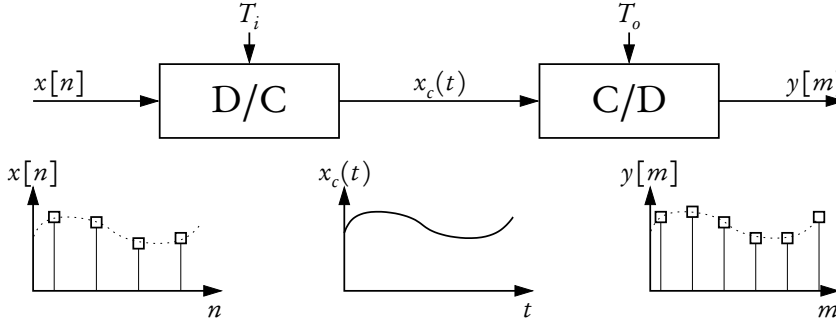
Figure 3.1: Reconstruction and resampling of continuous-time signals as conceptual model for delaying discrete-time sequences.

3.2 Sampling and Reconstruction of Continuous-Time Signals

Conceptually, delaying a discrete-time signal can be modeled by a conversion into a continuous-time signal followed by a continuous-to-discrete-time conversion. Moreover, the most important effects that occur in practical, discrete-time implementations of time-varying delay algorithms are best described in terms of this hybrid continuous/discrete-time representation. Because the theory of sampling and reconstruction of continuous-time signals is treated in detail in numerous monographs on general digital signal processing and multirate systems (e.g. [OSB99, CR83, Hen02]), only the notation and the basic concepts are introduced here.

The signal flow of an arbitrary delay operation by reconstruction and sampling is depicted in Figure 3.1. A discrete-time sequence $x[n]$ with sampling frequency f_i or, equivalently, sampling period T_i, is converted to a continuous-time signal $x_c(t)$ by means of a discrete-to-continuous-time converter D/C. The signal is resampled by a continuous-to-discrete-time conversion C/D at a set of desired output instants, resulting in the sequence $y[m]$. In this model, the output instants are symbolized by the period T_o of the output sample clock. Depending on the application, this output sample clock might represent different operations. For sample rate conversion operations, T_o differs from T_i in general. For delay operations, the output instants are determined by the times associated with the input samples minus a given delay value.

In the following, models for the C/D and D/C conversion processes are described. It is worth noting that the order of representation is reversed compared to the sequence in which these operations appear in the signal flow of the delay operation. There are two main reasons for this decision. First, starting from sampling introduces the concepts in a more natural way and is consistent with most treatises on the subject. Second and more important, the properties of the continuous-time signal $x_c(t)$ form the main focus of attention. Therefore, is is reasonable to start from this continuous-time representation.

Furthermore, it is worth emphasizing that a discrete-time sequence is not required to be generated by a sampling process [OSB99]. Thus, a continuous-time signal corresponding to this sequence does not necessarily exist. However, operations such as arbitrary delay or resampling require an unambiguous representation for signal values between sampling instants. In this way,

(a) Continuous-time signal $x_c(t)$

(b) Dirac impulse train $s(t)$

(c) Pulse amplitude modulated signal $x_s(t)$

(d) Discrete-time sequence $x[n]$
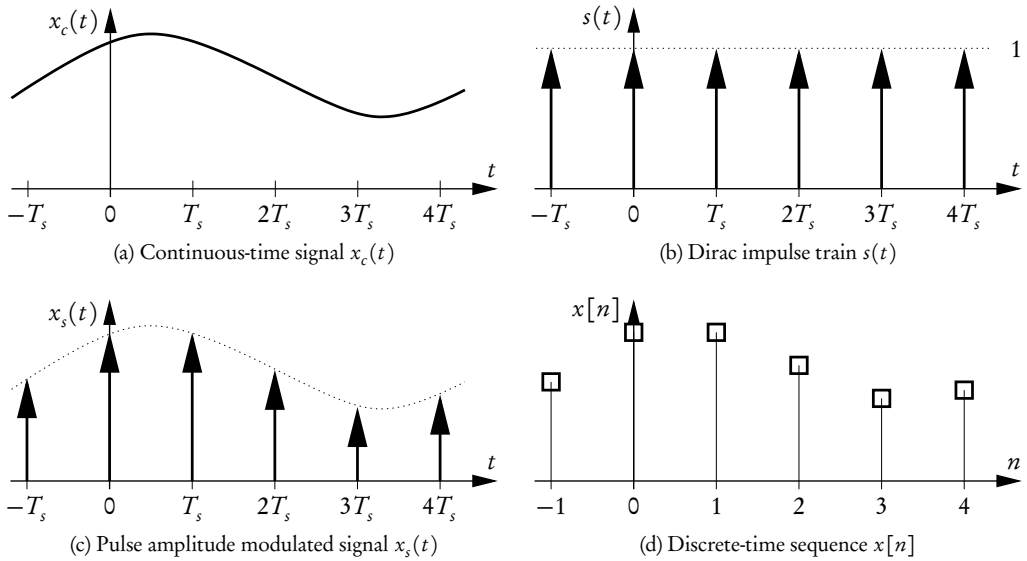
Figure 3.2: Conversion of a continuous-time signal to a discrete-time sequence.

the representation based on sampling and reconstruction of bandlimited continuous-time signals is an appropriate way to model these operations.

3.2.1 Continuous-to-Discrete-Time Conversion

The transformation of a continuous-time signal $x_c(t)$ to a discrete-time sequence $x[n]$ with sampling frequency f_s is modeled as a two-step process: In the first step, the function is converted to a modulated pulse train, while in the second step, these pulses are taken as the elements of the discrete-time sequence. The signals involved in this process are depicted in Figure 3.2.

Figure 3.2a shows a continuous-time signal $x_c(t)$. Modulation, that is multiplication, with a Dirac impulse train $s(t)$ with period $T_s = \frac{1}{f_s}$ yields the pulse amplitude modulated (PAM) signal $x_s(t)$ [CR83]. Here, T_s and f_s denote the sampling period and the sampling frequency, respectively. The periodic impulse train $s(t)$ is defined as

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT_s), \tag{3.1}$$

where $\delta(t)$ denotes the continuous-time Dirac impulse. The Dirac impulse is a distribution or generalized function that can be defined, for instance, as the limit of a delta sequence or as a measure. Informally, the Dirac impulse $\delta(t)$ forms an infinitesimally narrow impulse with area 1 concentrated at $t = 0$. This is embodied by the defining property [Bra00]

$$\int_{-\infty}^{\infty} f(t)\delta(t)\,\mathrm{d}t = f(0). \tag{3.2}$$

However, this formula must be interpreted in a symbolic way only, because it does not represent a proper integral either in the Riemann or in the Lebesgue sense. Applying a shift to the argument of $\delta(t)$ results in the so-called sifting property of the Dirac impulse [Bra00, OSB99]

$$f(\tau) = \int_{-\infty}^{\infty} f(t)\delta(t - \tau)\,\mathrm{d}t\,. \tag{3.3}$$

That is, the function value $f(\tau)$ for an arbitrary argument τ can be determined using an appropriately shifted Dirac impulse. The impulse train $s(t)$ and the pulse-modulated signal $x_s(t)$ are depicted in Figures 3.2b and 3.2c, respectively. The pulse amplitude modulated signal $x_s(t)$ is expressed as

$$x_s(t) = x_c(t)s(t) \tag{3.4a}$$

$$= x_c(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \tag{3.4b}$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s). \tag{3.4c}$$

Formally, the sample value $x[n]$ is determined from the area of the impulse of $x_s(t)$ at $t = nT_s$

$$x[n] = \lim_{\varepsilon \to 0} \int_{nT_s - \varepsilon}^{nT_s + \varepsilon} x_c(t)s(t)\,\mathrm{d}t\,. \tag{3.5}$$

According to the sifting property of the Dirac impulse (3.3), it follows that

$$x[n] = x_c(nT_s), \tag{3.6}$$

that is, the discrete-time sequence is determined by the values of the continuous signal at the sampling instants.

To analyze the frequency-domain properties of the sampling process, a continuous-time Fourier transform is applied to $x_s(t)$. The Fourier transform operator and the inverse Fourier transform for angular frequency variables are defined as

$$F(\Omega) = \quad \mathscr{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t)\mathrm{e}^{-\mathrm{j}\Omega t}\,\mathrm{d}\Omega \tag{3.7a}$$

$$f(t) = \mathscr{F}^{-1}\{F(\Omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\Omega)\mathrm{e}^{\mathrm{j}\Omega t}\,\mathrm{d}t\,. \tag{3.7b}$$

The continuous-time Fourier transform $S(\mathrm{j}\Omega) = \mathscr{F}\{s(t)\}$ of the Dirac impulse train $s(t)$ is also a periodic impulse train

$$S(\mathrm{j}\Omega) = \sum_{n=-\infty}^{\infty} \delta\left(\Omega - k\frac{2\pi}{T_s}\right) = \Omega_s \sum_{n=-\infty}^{\infty} \delta(\Omega - k\Omega_s). \tag{3.8}$$

Here, Ω_s denotes the angular sampling frequency

$$\Omega_s = 2\pi f_s = \frac{2\pi}{T_s}. \tag{3.9}$$

Modulating, that is multiplying, the input signal $x_c(t)$ with the Dirac impulse train $s(t)$ in the time domain according to (3.4) corresponds to a convolution of the Fourier transforms $X_c(j\Omega)$ and $S(j\Omega)$

$$X_s(j\Omega) = \frac{1}{2\pi} X_c(j\Omega) * S(j\Omega), \tag{3.10}$$

where $*$ denotes the continuous convolution operator

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(\tau) g(x - \tau) d\tau. \tag{3.11}$$

The scaling factor $\frac{1}{2\pi}$ in (3.10) results from the use of angular frequency variables in the Fourier transform operator (3.7a). Applying (3.8) to (3.10) yields

$$X_s(j\Omega) = \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X_c\left(j[\Omega - l\Omega_s]\right). \tag{3.12}$$

That is, the Fourier transform of $X_s(j\Omega)$ consists of an infinite number of replications of $X_c(j\Omega)$ scaled by $1/T_s$ and shifted by multiples of the period Ω_s.

An alternative form for $X_s(j\Omega)$ is obtained by applying the continuous Fourier transform operator (3.7a) to the time-domain definition of $x_s(t)$ (3.4c) utilizing the relation to the samples $x[n]$ (3.6)

$$X_s(j\Omega) = \mathscr{F}\left\{\sum_{-\infty}^{\infty} x_c(nT_s)\delta(t - nT_s)\right\}$$

$$= \sum_{-\infty}^{\infty} x_c(nT_s) e^{-j\Omega T_s n} \tag{3.13a}$$

$$= \sum_{-\infty}^{\infty} x[n] e^{-j\Omega T_s n}. \tag{3.13b}$$

The frequency-domain properties of the discrete-time sequence $x[n]$ are described by its discrete-time Fourier transform (DTFT) $X(e^{j\omega})$. The transform pair of the DTFT is

$$X(e^{j\omega}) = \text{DTFT}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n} \tag{3.14a}$$

$$x[n] = \text{DTFT}^{-1}\left\{X(e^{j\omega})\right\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega. \tag{3.14b}$$

Here, ω denotes the normalized angular frequency

$$\omega = 2\pi \frac{\Omega}{\Omega_s} = \Omega T_s .$$

(3.15)

The discrete-time Fourier transform of the sequence $x[n]$ is obtained by formal application of the discrete-time Fourier transform operator (3.14a)

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \sum_{n=-\infty}^{\infty} x[n]\mathrm{e}^{-\mathrm{j}\omega n} .$$

(3.16)

Comparing (3.16) to (3.13b) shows that $X(\mathrm{e}^{\mathrm{j}\omega})$ is identical to $X_s(\mathrm{j}\Omega)$ if the normalized angular frequency variable ω equals ΩT_s

$$X(\mathrm{e}^{\mathrm{j}\omega}) = X_s(\mathrm{j}\Omega)\big|_{\omega=\Omega T_s} = X_s\left(\mathrm{j}\frac{\omega}{T_s}\right) .$$

(3.17)

Applying (3.12) to (3.17) results in

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X_c\left(\mathrm{j}\left[\frac{\omega}{T_s} - l\frac{2\pi}{T_s}\right]\right) .$$

(3.18)

Thus, the discrete-time Fourier transform $X(\mathrm{e}^{\mathrm{j}\omega})$ consists of replications of the Fourier transform of $x_c(t)$ spaced in frequency with period $\Omega_s = 2\pi/T_s$. This explicit relation reveals a necessary condition that must hold if $x[n]$ shall represent the continuous signal $x_c(t)$ uniquely. This requirement is embodied in the Nyquist sampling theorem [Sha49, Jer77, Uns00, Mei02, OSB99]:

> If a function $f(t)$ contains no frequencies higher than W cps [cycles per seconds, i.e. Hertz (Hz)], it is completely determined by giving its ordinates at a series of points spaced $1/2W$ apart.

A variety of different names exists for this theorem due to its numerous inventors, including Whittaker, Nyquist, Shannon and Kotel'nikov. For a discussion, see [Mei02, p. 326]. Here, the naming from [OSB99] is adopted.

Formally, this condition is expressed as

$$X_c(\mathrm{j}\Omega) = 0 \quad \text{for } |\Omega| \geq \frac{\Omega_s}{2} .$$

(3.19)

If (3.19) is fulfilled, the discrete-time Fourier transform $X\left(\mathrm{e}^{\mathrm{j}\omega}\right)$ (3.18) simplifies to

$$X(\mathrm{e}^{\mathrm{j}\omega}) = \frac{1}{T_s} X_c(\mathrm{j}\Omega) = \frac{1}{T_s} X_c\left(\mathrm{j}\frac{\omega}{T_s}\right) .$$

(3.20)

The sampling theorem can be illustrated using the Fourier transform of the pulse amplitude modulated signal $X_s(\mathrm{j}\Omega)$ (3.12). Figure 3.3a shows the Fourier transform of a continuous signal

(a) Continuous-time signal $X_c(j\Omega)$

(b) Dirac impulse train $S(j\Omega)$

(c) Pulse amplitude modulated signal $X_s(j\Omega)$, $\Omega_N < \frac{\Omega_s}{2}$ (no aliasing)

(d) Pulse amplitude modulated signal $X_s(j\Omega)$, $\Omega_N \geq \frac{\Omega_s}{2}$ (aliasing)
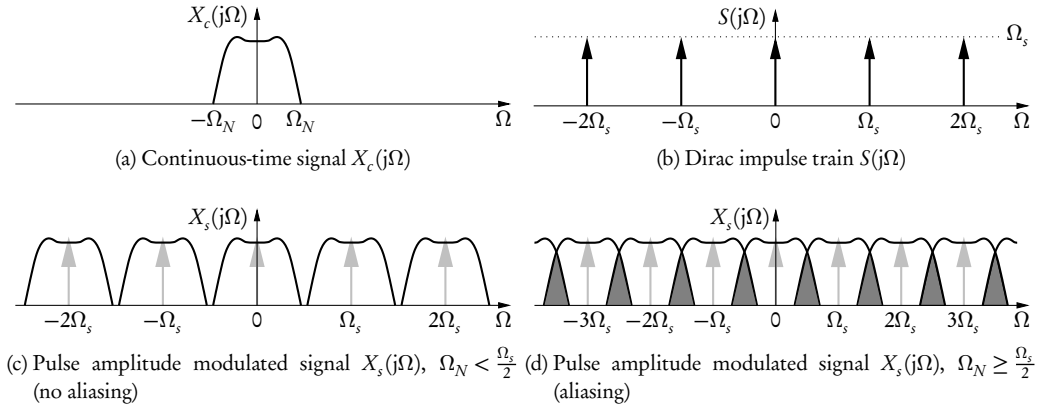
Figure 3.3: Frequency-domain interpretation of the sampling process.

that is bandlimited to Ω_N, where Ω_N is commonly referred to as the Nyquist frequency [OSB99] that is defined by

$$X_c(j\Omega) = 0 \quad \text{for } |\Omega| \geq \Omega_N. \tag{3.21}$$

The Fourier transform of a periodic impulse train with sampling frequency Ω_s is shown in Figure 3.3b. Figures 3.3c and 3.3d depict the pulse amplitude modulated signals for two different sampling frequencies. In 3.3c, the sampling theorem holds, that is $\Omega_s > 2\Omega_N$. Consequently, the spectral replications of $X_c(j\Omega)$ do not overlap. In contrast, Ω_N exceeds half the sampling frequency in Figure 3.3d. Therefore, consecutive replications of $X_c(j\Omega)$ overlap. This phenomenon is commonly denoted as *aliasing* or *spectral foldover*. In this way, different frequency components of the input signal are mapped or aliased to the same frequency in the pulse amplitude modulated signal and consequently to the discrete-time sequence $x[n]$. Because different signal components mapped to the same frequency are indistinguishable, aliasing cannot be compensated by subsequent analog or discrete-time signal processing. Thus, aliasing is an irreversible process [Hen02].

So, in order to prevent aliasing, the input signal to the continuous-to-discrete-time conversion must be bandlimited to $\Omega_s/2$. If the signal $x_c(t)$ does not meet this requirement inherently, a continuous-time prefilter or *anti-aliasing filter* $H_{aa}(j\Omega)$ [OSB99] is commonly applied, resulting in the signal flow graph shown in Figure 3.4. The input signal is filtered by an anti-aliasing filter with impulse response $h_{aa}(t)$, yielding the bandlimited signal $x_a(t)$ which is used as input into the C/D converter. The ideal anti-aliasing filter $\hat{H}_{aa}(j\Omega)$ is a continuous-time lowpass filter with
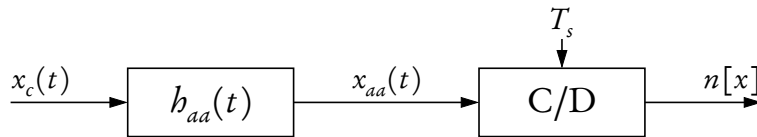
Figure 3.4: Continuous-to-discrete-time conversion with explicit anti-aliasing filter.

(a) Bandlimited continuous-time signal corresponding to $X(e^{j\omega})$

(b) Pulse amplitude modulated signal $X_s(j\Omega)$

(c) Ideal reconstruction filter $\widehat{H}_r(j\Omega)$

(d) Reconstructed continuous-time signal $X_r(j\Omega)$
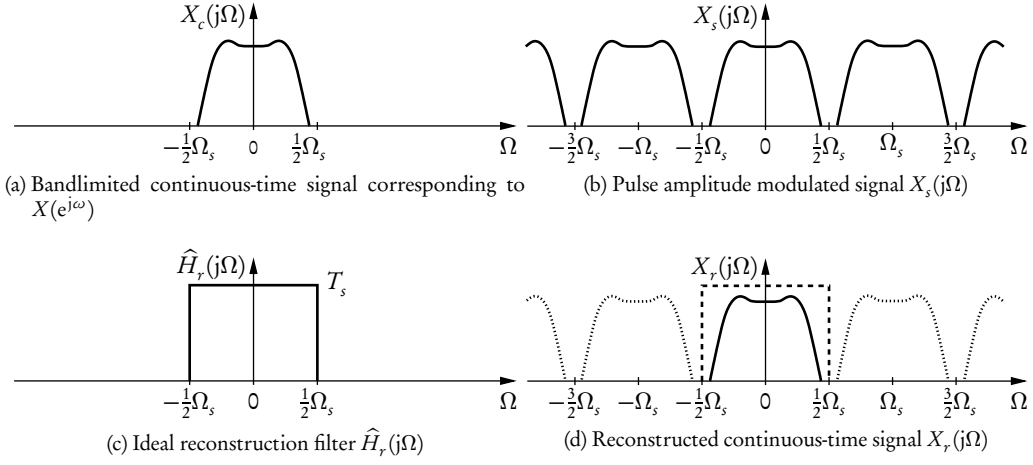
Figure 3.5: Frequency-domain interpretation of the reconstruction of a continuous-time signal from a discrete sequence.

cutoff frequency $\Omega_s/2$

$$\widehat{H}_{aa}(j\Omega) = \begin{cases} 1, & |\Omega| < \frac{\Omega_s}{2} \\ 0, & |\Omega| \geq \frac{\Omega_s}{2} \end{cases}. \tag{3.22}$$

3.2.2 Reconstructing a Continuous-Time Signal from a Discrete Sequence

Alike sampling, reconstruction of a continuous-time signal from a discrete sequence is modeled as a two-step process. First, the sequence is transformed to a pulse modulated continuous-time signal. Subsequent filtering with a reconstruction filter yields the reconstructed continuous-time signal $x_r(t)$. In Figure 3.5, the reconstruction process is depicted in the frequency domain.

Modulating the sequence $x[n]$ with an impulse train $s(t)$ (3.1) yields a continuous-time pulse amplitude modulated signal $x_s(t)$ as introduced in (3.4)

$$x_s(t) = \sum_{n=-\infty}^{\infty} x[n]\delta\left(t - nT_s\right) \tag{3.23a}$$

$$= \sum_{n=-\infty}^{\infty} x_c(nT_s)\delta\left(t - nT_s\right). \tag{3.23b}$$

For the latter form (3.23b), it is assumed that $x[n]$ corresponds unambiguously to a continuous-time signal $x_c(t)$ sampled with T_s, implying that the Nyquist criterion (3.19) is fulfilled. As stated above, it is not required in general that a discrete-time sequence has been obtained by sampling a bandlimited function. However, the Nyquist condition with respect to the output

sampling frequency forms a necessary condition for an exact and unambiguous reconstruction of a continuous-time signal from a discrete-time sequence.

Figure 3.5a shows the spectrum of a bandlimited signal $x_c(t)$ corresponding to the sequence $x[n]$. The Fourier transform $X_s(j\Omega)$ of the pulse amplitude modulated signal (3.12)

$$X_s(j\Omega) = \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X_c\left(j[\Omega - l\Omega_s]\right) \tag{3.24}$$

consists of an infinite number of spectral replications or images of $X_c(j\Omega)$ as depicted in Figure 3.5b. To recover the signal $X_c(j\Omega)$, all replications except the one centered at $\Omega = 0$ have to be removed. This process is modeled by filtering $X_s(j\Omega)$ with a continuous-time lowpass filter $H_r(j\Omega)$, which is commonly referred to as the *reconstruction filter*. The frequency response of the ideal reconstruction filter $\widehat{H}_r(j\Omega)$, making no assumptions about the frequency contents of $X_c(j\Omega)$ further than the Nyquist criterion, is an ideal lowpass with cutoff frequency $\Omega_s/2$

$$\widehat{H}_r(j\Omega) = \begin{cases} T_s, & |\Omega| < \frac{\Omega_s}{2} \\ 0, & |\Omega| \geq \frac{\Omega_s}{2} \end{cases} . \tag{3.25}$$

The passband gain of $T_s = 2\pi/\Omega_s$ [OSB99] compensates the amplitude scaling in the PAM signal (3.24), which is a consequence of the frequency normalization performed in the sampling process. Applying the ideal reconstruction filter to $x_s(t)$ yields the reconstructed signal $x_r(t)$ with Fourier transform

$$X_r(j\Omega) = \widehat{H}_r(j\Omega) \frac{1}{T_s} \sum_{l=-\infty}^{\infty} X_c\left(j\left[\Omega - l\Omega_s\right]\right) \tag{3.26a}$$

$$= X_c(j\Omega) = X\left(e^{j\omega}\right)\Big|_{\omega = \frac{2\pi\Omega}{\Omega_s}} . \tag{3.26b}$$

In this way, the spectrum of the reconstructed signal is identical to the spectrum of the bandlimited continuous-time signal $X_c(j\omega)$ represented by the discrete-time Fourier transform $X(e^{j\omega})$ of the sequence $x[n]$. This operation is illustrated in Figure 3.5d.

In the time domain, ideal reconstruction implies that the reconstructed signal is identical to the bandlimited signal $x_c(t)$

$$x_r(t) = x_c(t). \tag{3.27}$$

The frequency-domain filtering operation (3.26a) corresponds to a convolution with the impulse response of the ideal reconstruction filter $\widehat{h}_r(t)$

$$\widehat{h}_r(t) = \mathscr{F}^{-1}\left\{\widehat{H}_r(j\Omega)\right\} \tag{3.28a}$$

$$= \frac{1}{2\pi} \int_{-\frac{\Omega_s}{2}}^{\frac{\Omega_s}{2}} T_s e^{j\Omega t} \, d\Omega \quad \text{with } \Omega_s = \frac{2\pi}{T_s} \tag{3.28b}$$

(a) Impulse response of the ideal reconstruction filter

(b) Reconstructed signal
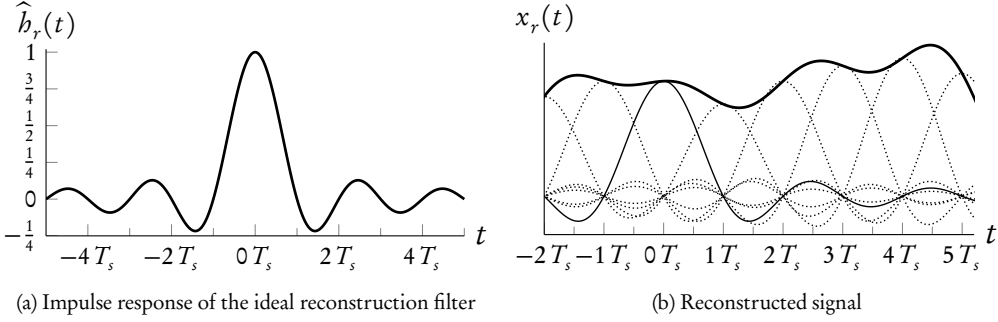
Figure 3.6: Reconstruction of a continuous-time signal from a discrete-time sequence.

$$= \frac{\sin\left(\frac{\pi}{T_s}t\right)}{\frac{\pi}{T_s}t} \tag{3.28c}$$

$$= \mathrm{sinc}\left(\frac{\pi}{T_s}t\right). \tag{3.28d}$$

Here, $\mathrm{sinc}(\cdot)$ denotes the unnormalized Sinc or cardinal sine function [GS90]

$$\mathrm{sinc}(x) = \begin{cases} \frac{\sin x}{x}, & x \neq 0 \\ 1, & x = 0 \end{cases}. \tag{3.29}$$

Thus, filtering with the ideal reconstruction filter $\widehat{h}_r(t)$ yields

$$x_r(t) = x_s(t) * \widehat{h}_r(t) \tag{3.30a}$$

$$= \int_{-\infty}^{\infty} x_c(\tau)\delta\left(\tau - nT_s\right)\widehat{h}_r(t-\tau)\mathrm{d}\tau \tag{3.30b}$$

$$= \sum_{n=-\infty}^{\infty} x[n]\widehat{h}_r(t-nT_s). \tag{3.30c}$$

In this way, ideal reconstruction at an arbitrary time t corresponds to a discrete convolution of the sequence $x[n]$ with a discrete sequence obtained by sampling the continuous impulse response $\widehat{h}_r(t)$ at the instants $t - nT_s$. Figure 3.6a depicts the ideal reconstruction filter. Likewise, reconstruction as a superposition of shifted and scaled sinc functions is illustrated in Figure 3.6b.

The reconstruction formula (3.30) has been first stated by E. T. Whittaker in 1915 [Whi15, Whi29] and is commonly denoted as the *cardinal series* or the *cardinal interpolation formula*

$$x_r(t) = \sum_{n=-\infty}^{\infty} x[n]\widehat{h}_r(t-nT_s) \tag{3.31a}$$

$$= \sum_{n=-\infty}^{\infty} x[n] \frac{\sin\left(\frac{\pi}{T_s}\left(t - nT_S\right)\right)}{\frac{\pi}{T_s}\left(t - nT_S\right)} \,. \tag{3.31b}$$

The cardinal series (3.31b) has a number of interesting properties. First, $\widehat{h}_r(t)$ is an analytic function [Sch46a]. In particular, it is infinitely differentiable and has an unbounded support. Second, since

$$\widehat{h}_r(t) = \begin{cases} 1, & t = 0 \\ 0, & t = \pm T_s, \pm 2T_s, \pm 3T_s, \dots \end{cases}, \tag{3.32}$$

the interpolation function is exact at the sampling instants. That is, it fulfills

$$x_r(nT_s) = x[n] \quad \text{for } n \in \mathbb{Z}, \tag{3.33}$$

which is commonly referred to as the *interpolation condition*.

On the other hand, $\widehat{h}_r(t)$ dampens out proportionally to $^1/_t$ as t approaches $\pm\infty$. As stated in [Sch46a], "however, its excessively slow rate of damping, for increasing x, makes the classical cardinal series [...] inadequate for numerical purposes".

Apart from this property, the cardinal series is not applicable for practical implementations of resampling, as it involves an infinite number of operations and cannot be implemented in causal form by introducing a finite delay. Moreover, it is not bounded-input-bounded-output (BIBO) stable as the sampled sinc function is not absolutely summable [Väl95a]. For these reason, only approximations of the ideal reconstruction filter are used in actual implementations. These approximations typically introduce errors into the reconstruction process which fall into different categories. First, the passband region of the reconstruction filter might be subject to deviations. Second, phase errors are potentially introduced. However, this class of errors can be avoided by reconstruction filters with symmetric impulse responses, which are made causal by a finite shift. Insufficient attenuation of the spectral replications of $X_s(j\Omega)$ in the stopband region of the reconstruction filter $H_r(j\Omega)$ (3.25) forms the third class of errors, which are commonly referred to as *imaging errors*.

3.2.3 Discussion

In this section, the processes of sampling and reconstructing continuous-time signals have been reviewed. Apart from idealized, abstract operations such as modulation with impulse trains, two continuous-time filters form the core of these operations. Their characteristics affect the errors of sampling and reconstruction directly. The anti-aliasing filter $H_{aa}(j\Omega)$ is required to limit the bandwidth of the continuous-time input signal such that the Nyquist criterion is fulfilled in the C/D conversion. Defects in the characteristics of this filter give rise to aliasing artifacts in the sampled sequence, which cannot be corrected by subsequent operations. The reconstruction or anti-imaging filter $H_r(j\Omega)$ is required to suppress the signal images that result from the conversion of a discrete-time sequence into a continuous-time signal. Incomplete attenuation of these images results in imaging artifacts.

3.3 Fractional Delay Filtering

Fractional Delay (FD) filtering denotes the process of delaying a discrete-time sequence by an arbitrary amount of time [LVKL96, KPPV07]. Delaying a sequence by an integer multiple of the sample period of the discrete-time systems can be performed by shifting the index of the sequence and is therefore an exact operation that does not require arithmetic operations. In contrast, delay values that correspond to time values between the sampling times of the input sequence require sophisticated algorithms.

FD filtering is utilized in many applications of signal processing, e.g. timing recovery in digital modems [EGH93], digital differentiators [Tse05], software-defined radio [DJR10, CY03] or comb filter design [PT98].

In the field of audio signal processing, applications of FD filtering include digital audio effects [Dat97, Zöl02], physical sound synthesis [Väl95a, Smi10a, VPEK06] or sound reproduction systems [Str00, FGKS07].

Additionally, FD filtering techniques are frequently used for sample rate conversion, e.g. [Har97, TKC94]. However, it should be emphasized that, although FD filtering and SRC are closely related, they also exhibit fundamental differences. These requirements of sample rate conversion systems are described in Section 3.4, and the distinctions will be pointed out in more detail in Section 3.9.

3.3.1 Definition

A fractional delay (FD) filter transforms a discrete-time sequence $x[n]$ into a discrete-time sequence $y[n, D]$ which approximates $x[n]$ delayed by a time delay D given in samples. In general, D is not an integer number.

Fractional delay filtering is best defined in the framework of sampling and reconstruction of discrete-time signals described in Section 3.2. For a given sampling frequency f_s or sampling period $T_s = 1/f_s$, the sequence $x[n]$ corresponds to a continuous-time signal $x_c(t)$. If $x_c(t)$ is bandlimited to the cutoff frequency $f_c = f_s/2 = 1/(2T_s)$ or equivalently $\Omega_c = \Omega_s/2 = \pi/T_s$, the Nyquist criterion (3.19) holds, and thus $x_c(t)$ is uniquely defined by the samples $x[n]$. In this case, the sequence $y[n, D]$ can be obtained exactly and uniquely by reconstructing the continuous-time signal $x_c(t)$ followed by evaluation at the desired output instants.

$$y[n, D] = x_c\left(nT_s - \tau\right) \tag{3.34a}$$
$$= x_c\left(nT_s - DT_s\right) \tag{3.34b}$$

Here $\tau = DT_s$ denotes the time delay (in seconds) corresponding to the sample delay D. Application of the reconstruction formula (3.31) to (3.34) yields

$$y[n, D] = x_c\left([n - D] T_s\right) \tag{3.35a}$$
$$= \sum_{l=-\infty}^{\infty} x[l] \operatorname{sinc}\left(\frac{\pi}{T_s}[n - D - l] T_s\right) \tag{3.35b}$$
$$= \sum_{l=-\infty}^{\infty} x[l] \operatorname{sinc}\left(\pi[n - l - D]\right) \tag{3.35c}$$

$$= \sum_{l=-\infty}^{\infty} x[n-l] \operatorname{sinc}(\pi[l-D]) \tag{3.35d}$$

$$= x[n] * \widehat{h}(n,D). \tag{3.35e}$$

The equivalence between (3.35c) and (3.35d) is a consequence of the commutativity of the discrete convolution operation [Bra00, OSB99]

$$f * g = g * f \tag{3.36a}$$

$$\sum_{m=-\infty}^{\infty} f[m]g[n-m] = \sum_{m=-\infty}^{\infty} f[n-m]g[m]. \tag{3.36b}$$

In this way, the ideal fractional delay operation corresponds to a discrete convolution with the impulse response of the ideal fractional delay filter $\widehat{h}(n,D)$

$$\widehat{h}(n,D) = \operatorname{sinc}(\pi(n-D)) \tag{3.37}$$

which is obtained by shifting and sampling the ideal reconstruction filter $\widehat{h}_r(t)$ (3.28) for a sampling period normalized to $T_s = 1$. From the representation (3.35d), it is apparent that a fractional delay operation is generally implemented as a discrete-time filter.

The frequency response of the ideal FD element follows from the frequency-domain representation of a time shift

$$\widehat{H}\left(e^{j\omega}, D\right) = \operatorname{DTFT}\left\{\widehat{h}(n,D)\right\} = e^{-j\omega D}. \tag{3.38}$$

3.3.2 Performance Measures for Realizable FD filters

As argued in Section 3.2.2, system responses based on the ideal reconstruction filter $\widehat{h}_r(t)$ (3.28) are not realizable and thus not applicable for practical implementations. Consequently, an approximation of the response of the ideal FD element by a realizable filter structure invariably introduces approximation errors. To assess these inaccuracies, several performance measures are in use. The expressiveness of these measures depends on the application. For a more elaborate discussion, see [LVKL96].

The complex error function is defined as the difference between the complex frequency responses of the FD filter $H(e^{j\omega}, D)$ and the ideal FD element [Väl95a, LVKL96]

$$E(e^{j\omega}, D) = H(e^{j\omega}, D) - \widehat{H}(e^{j\omega}, D) \tag{3.39a}$$

$$= H(e^{j\omega}, D) - e^{-j\omega D}. \tag{3.39b}$$

As for general discrete-time filters, the frequency response of a FD element can be partitioned into a magnitude response $|H(e^{j\omega}, D)|$ and a phase response $\Theta(\omega)$

$$H(e^{j\omega}, D) = |H(e^{j\omega}, D)|\Theta(\omega). \tag{3.40}$$

The magnitude response $\widehat{H}(e^{j\omega}, D)$ of the ideal FD element is 1, that is, it represents an allpass system

$$\left|\widehat{H}\left(e^{j\omega}, D\right)\right| = 1. \tag{3.41}$$

The phase response of discrete-time filter is the phase angle of the complex frequency response [OSB99, Smi10b]

$$\Theta(\omega) = \sphericalangle H\left(\mathrm{e}^{\mathrm{j}\omega}, D\right) . \tag{3.42}$$

In the strict sense, the phase angle of $H(\mathrm{e}^{\mathrm{j}\omega}, D)$ is restricted to the interval $(-\pi, \pi]$. For this reason, $\Theta(\omega)$ is used to denote the continuous [OSB99] or unwrapped phase [Smi10b] which is a continuous function of frequency. The phase response of the ideal FD filter is

$$\widehat{\Theta}(\omega) = \sphericalangle \mathrm{e}^{-\mathrm{j}\omega D} \tag{3.43a}$$
$$= -D\omega . \tag{3.43b}$$

Consequently, the ideal phase response is a linear function of the frequency, and thus $\widehat{H}(\mathrm{e}^{\mathrm{j}\omega}, D)$ forms a linear-phase system.

Because the utility of the phase response measure for FD filters is often limited [LVKL96], two other measures are used to describe the phase behavior of a system: the phase delay $\tau_p(\omega)$ and the group delay $\tau_g(\omega)$

$$\tau_p(\omega) = -\frac{\Theta(\omega)}{\omega} \tag{3.44}$$

$$\tau_g(\omega) = -\frac{d}{d\omega}\Theta(\omega). \tag{3.45}$$

For the ideal FD element, group and phase delay are given by

$$\widehat{\tau}_p(\omega) = D \tag{3.46a}$$
$$\widehat{\tau}_g(\omega) = D . \tag{3.46b}$$

In this way, $\widehat{\tau}_p(\omega)$ and $\widehat{\tau}_g(\omega)$ are identical and independent of frequency, which is a direct consequence of the linear-phase property of $\widehat{H}\left(\mathrm{e}^{\mathrm{j}\omega}, D\right)$. Moreover, they equal the desired fractional delay value D.

In contrast, realizable FD filters do not exhibit linear-phase behavior except for particular values of D. While IIR filters generally exhibit a nonlinear phase, FIR FD filters may attain linear phase only if the delay is an integer number of samples or if the delay lies halfway between two integers [Sar93, OSB99].

The different performance measures are shown in Figure 3.7 for an exemplary FD filter based on Lagrange interpolation of order $N = 3$. It is observed that the error typically depends on the delay parameter D, and that the response of the ideal FD is assumed only for specific values of D.

3.3.3 Synthesizing Arbitrary Delay Values

In its original form, a FD filter is designed for a single, specific delay value D. However, many applications require different delays to be synthesized, which might be changed at run time.

(a) Amplitude response $|H(e^{j\omega}, D)|$

(b) Phase response $\sphericalangle H(e^{j\omega}, D)$

(c) Phase delay $\tau_p(\omega)$

(d) Group delay $\tau_p(\omega)$
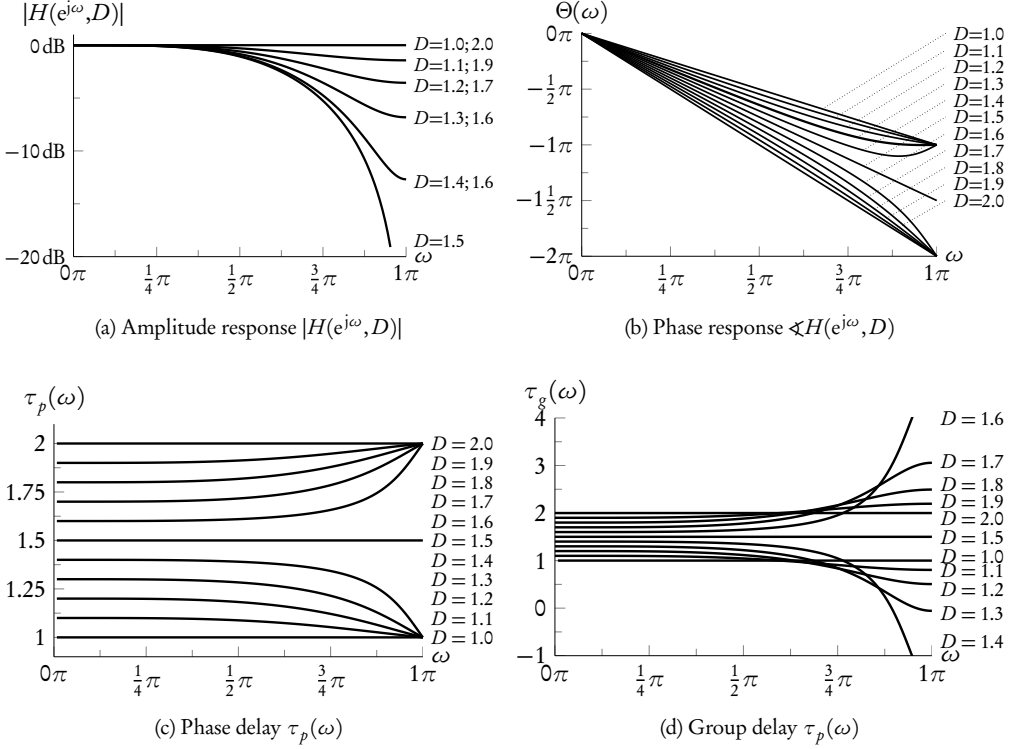
Figure 3.7: Performance measures for VFD filters. Example based on Lagrange interpolation, order $N = 3$.

Algorithms that enable such variable delays are commonly denoted as *variable fractional delay* (VFD) filters.

A first step to handle arbitrary delay values is to partition the delay D into an integer delay D_{int} and a fractional delay value d

$$D = D_{int} + d \qquad \text{with } d_{min} \le d \le d_{max}. \tag{3.47}$$

This transformation is necessary because VFD filters provide optimal quality only within a fixed interval $[d_{min}, d_{max}]$ of the delay value (e.g. [EGH93, Väl95a, LVKL96]). By partitioning the delay value according to (3.47), the VFD structure is always operated within this optimal range. At the same time, the VFD element must accept delay values only within a fixed, typically unit-sized range.

Despite their names, D_{int} and d do not necessarily represent the integer and fractional part of D. In fact, several fractional delay ranges are in use (see, among others, [Väl95a, LVKL96, VS96b, Den05])

$$0 \le d \le 1 \tag{3.48a}$$

$$-\frac{1}{2} \leq d \leq \frac{1}{2} \qquad\qquad (3.48b)$$

$$\frac{N-1}{2} \leq d \leq \frac{N+1}{2}, \qquad\qquad (3.48c)$$

where N denotes the filter order of the VFD structure. For each of these choices, the integer part is determined such that (3.47) holds. It is noted that the fractional delay ranges (3.48) are deliberately defined as closed intervals. In this way, (3.47) becomes ambiguous at the end points of the interval. Resolving this ambiguity is left to the implementation, because it does not influence the analysis of a VFD element.

The fractional delay range affects the selection of the samples $x[n]$ involved in the calculation of a particular output sample $y[n, D]$. It also impacts the properties of a VFD structure. A uniform characterization of different fractional delay ranges, based on the notion of basepoint sets, has been performed in [Fra08] for VFD filtering algorithms based on Lagrange interpolation. In Section 3.5, this characterization is presented for the more general class of discrete-time resampling filters which includes VFD structures.

To facilitate arbitrary delay values within the fractional delay interval, three basic strategies are in use:

Storing a Set of Precomputed Coefficient Sets One way to enable variable FD filters is to store a set of FD filter coefficients designed for a dense grid of delay values in the fractional delay interval (3.48). At run time, the fractional delay value d is used for a table lookup, either using truncation or a nearest-neighbor selection, to determine the filter coefficients to be used in the FD filter. Research on the influence of the grid density is mainly focused on continuously changing fractional delay values and is consequently targeted at sampling rate conversion algorithms [VLSR00, DH98]. For fractional delay filters, the number of quantization steps must be chosen according to the admissible delay error, which depends on the application. If P denotes the number of stored coefficient sets for a unit-sized fractional delay interval, then the maximum delay error ΔD_{max}, which is the difference between the desired and the synthesized delay, is given by

$$\Delta D_{max} = \begin{cases} \frac{1}{2P}, & \text{for nearest-neighbor selection} \\ \frac{1}{P}, & \text{for selection by truncation} \end{cases}. \qquad (3.49)$$

It is noted that this limit does not account for the errors of realizable FD filters, but assumes that the stored coefficients represent ideal FD elements.

On modern processor architectures or hardware realizations, the memory required for storing the filter coefficients is only one of the drawbacks of this implementation. On such platforms, the efficiency is also reduced by the required memory bandwidth and the latency caused by lookup and loading of the filter coefficients.

Finally, several algorithms (e.g. [Har97]) have been proposed that perform interpolation between stored coefficients. However, it is difficult to categorize these algorithms. Typically, they exhibit features of different classes of algorithms, for instance structures based on oversampling or variable digital filters.

Explicit Calculation of the Filter Coefficients A second way to enable arbitrary delay values is to use FD filters whose coefficients are calculated during the run time of the system

using explicit, efficient formulas. In most cases, these filters are derived from numerical interpolation or approximation formulas such as Lagrange or spline interpolation, or represent simple IIR allpass designs such as Thiran filters [Thi71]. Because the calculation of the filter coefficients is separated from the actual filtering, it can be performed at a rate lower than the sampling frequency. This allows cost-efficient implementations if the delay value changes relatively seldom.

Variable Digital Filter Structures Variable Digital Filters (VDF) are discrete-time systems for filtering that are controlled by an additional control parameter. The term "variable digital filter" has been first coined in [SW70, SH82], although these works are mainly focused on filters with adjustable magnitude characteristics. A comprehensive overview of VDF is given in [SK97]. In the context of FD filtering, the use of variable filter structures has been introduced in [Far88], resulting in the so-called *Farrow structure*, which is used synonymously for VFD filtering based on variable digital filters.

Variable digital filter structures combine the calculation of the filter coefficients and the actual filtering into a single operation. In this way, VDF structures enable arbitrary delay values that might change arbitrarily often, that is, at every sampling instant, without increasing the computational effort.

3.3.4 Properties of Variable Fractional Delay Filters

In this section, some properties that are used to characterize variable fractional delay filters are introduced.

The Interpolation Condition In the context of sampling and reconstruction, the interpolation condition has been stated in (3.33). It represents the ability of a system to exactly reproduce the signal value at sampling instants. For FD filtering, this condition is expressed by

$$y[n, D] = x[n - D] \quad \text{for } D \in \mathbb{N} \tag{3.50}$$

that is, a fractional delay corresponding to an integer number of samples yields the same result as an index shift in the input sequence $x[n]$.

Several algorithms for FD filtering inherently fulfill the interpolation condition, for instance Lagrange interpolation (Section 3.6.3) or cardinal spline interpolation (Section 3.6.4.4). The design methods of some other variable FD filtering methods can be augmented by additional constraints to force the interpolation condition to be fulfilled.

Ultimately, the necessity of the interpolation condition depends on the application. If there are no stringent reasons to fulfill this constraint, the degrees of freedom in filter design that are required to meet the interpolation condition might be used advantageously to decrease the error with respect to a given metric over the whole fractional interval.

Errors Due to Delay Changes The change of the delay value during the run time of the system may result in different classes of errors.

- Since an instantaneous change of the delay value corresponds to a jump within the input waveform, such changes generally result in discontinuities of the output waveform. Strictly

speaking, this is not an error because it complies with the definition of FD filtering. However, for many applications, this effect is not desirable. For instance, in audio applications, discontinuities are highly audible due to their broad spectrum. For this reason, several methods have been proposed to mitigate the effects of delay changes. In [Väl95a], the segmentation of the delay change into a multitude of smaller changes is proposed while [Smi10a] suggests a crossfade between two signals delayed by the old and the new delay value, respectively. However, these methods generally require additional computational resources and often introduce additional errors, for instance comb filtering effects in case of a crossfade.

- If the filtering structure contains internal states that depend on previous filter coefficients, then a change of the delay value causes the FD filter to use intermediate values computed for a different delay value. This class of errors is termed *transient errors* [Väl95a]. In general, FD elements based on IIR structures are prone to this error, because IIR filters utilize past output samples. However, specific algorithms for FIR-based FD structures also exhibit transient errors if intermediate results dependent on the fractional delay value are stored internally. As in case of discontinuities, methods to decrease transient errors have been proposed (see, e.g. [Väl95a]). Again, these methods generally require additional computational effort and control logic.

3.3.5 Discrete-Time Structures for Fractional Delay Filters

In this section, it became clear that FD elements are generally implemented as discrete-time filtering processes. Actually, they are in general based on standard structures for discrete-time filtering, namely finite impulse response (FIR) or infinite impulse response (IIR) structures.

The advantages and disadvantages of these filters stated in the context of general DSP applications, e.g. [Sar93, OSB99], generally apply to FD filtering as well. Furthermore, some additional criteria specific to FD filtering have to be considered [LVKL96, Väl95a].

FIR filters enable a good approximation of the phase response. They are inherently stable, and the sensitivity to roundoff errors is generally low and does not cause instability [Sar93, OSB99]. Moreover, a number of very efficient design methods exists for FIR FD filters that enable the computation of the filter coefficients during the run time of the system. In addition, FIR filters do not contain internal states that depend on prior outputs of the system. This has two important implications. First, the data storage required for a FIR filter, that is, delay elements to hold past input samples, can be shared among multiple filters. Second, the filter characteristics can be changed during the run time of the system without introducing transient errors.

On the downside, FIR FD filters cannot provide exact allpass behavior for arbitrary delay values, thus invariantly causing amplitude errors with respect to the frequency response of the ideal FD filter (3.38). Moreover, FIR filters generally require a substantial higher filter order than an IIR filter to approximate a given design specification with a prescribed tolerance. As a direct implication, the implementation delay of a FIR filter is significantly higher than that of a comparable IIR filter.

In contrast, IIR FD filters can be implemented as allpass filters, thus representing the amplitude response of the ideal FD element exactly. On the contrary, the phase behavior of IIR filters is more difficult to control than in case of FIR filters. IIR filters are potentially subject to instability. They are more sensitive to finite wordlengths and roundoff errors, which may result in limit

Figure 3.8: General representation of a sample rate conversion system.

cycles. They are very few IIR FD filter design methods that enable an explicit calculation of the filter coefficients, Thiran filters [Thi71] being the most simple and most important filter class [LVKL96, Väl95a]. Finally, as IIR filters are recursive filters, they require past outputs to be stored. This complicates the control effort, especially if a large number of simultaneous filter operations is performed. Moreover, if the filter characteristics are changed during the run time of the system, the filter output depends on the previous filter coefficients, thus potentially causing transient phenomena. While methods to eliminate or attenuate such transients have been proposed (see e.g. [Väl95a]), these solutions are typically limited to a finite rate of discrete changes and require additional computational resources.

The main focus of this work is on arbitrary, time-varying delays that change the delay value relatively often or are continuously time-variant, and on efficient algorithms with predictable, preferably constant load. Subsuming the above characteristics, it becomes clear that FIR filters are far better suited to fulfill these requirements. For this reason, only VFD filters based on FIR structures, that is, finite discrete-time convolutions, are considered in the following.

3.4 Arbitrary Sample Rate Conversion

Sample rate conversion is a different approach to determine the value of a discrete-time sequence at arbitrary points in time. It denotes the transformation of a discrete-time sequence with a given sampling frequency to a discrete-time sequence having a different sampling frequency [CR81]. The general signal flow of a sampling rate conversion system is shown in Figure 3.8 [CR83, Hen02, Eva00b]: A discrete-time sequence $x[n] = x(nT_i)$ sampled with the input sampling period T_i or, correspondingly, angular sampling frequency Ω_i is transformed into a discrete-time sequence $y[m] = y(mT_o)$ sampled with the output period T_o or angular sampling frequency Ω_o. The notations $x(nT_i)$ and $y(mT_o)$ are common in the field of SRC, e.g. [Eva00b, Eva03, Hen02], to indicate that the sequences correspond to the sampling periods T_i and T_o, respectively. At the same time, this notation emphasizes the relation between the bandlimited continuous-time signals $x(t)$ and $y(t)$ and their sampled discrete-time representations. The sampling periods and frequencies are related by

$$\Omega_i = \frac{2\pi}{T_i} = 2\pi f_i \tag{3.51a}$$

$$\Omega_o = \frac{2\pi}{T_o} = 2\pi f_o, \tag{3.51b}$$

where f_i and f_o represent the ordinary sampling frequencies. It is noted that this general model of sample rate version assumes fixed input and output sampling frequencies, although this restriction might be loosened to support slowly time-varying input and output sampling rates [Eva00b, Eva03].
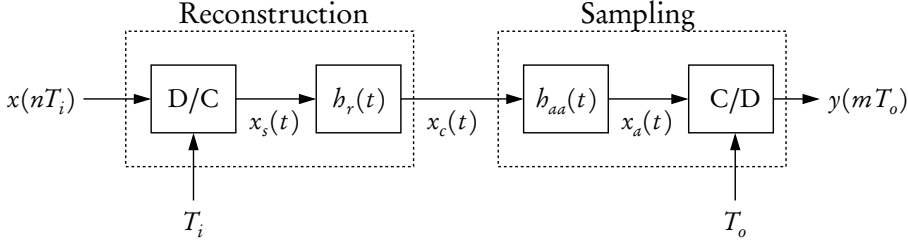
Figure 3.9: Sample rate conversion modeled by analog resampling.

The ratio between the between output and input sampling frequency is referred to as the *conversion ratio* or the SRC factor (e.g. [BLR03, LR03])

$$R = \frac{\Omega_o}{\Omega_i} = \frac{T_i}{T_o}. \tag{3.52}$$

3.4.1 Sample Rate Conversion as an Analog Resampling Process

Sample rate conversion can be conceptually modeled as an analog resampling process: The sequence $x(nT_i)$ is transformed to a continuous-time signal $x_c(t)$ followed by resampling at the output rate T_o to obtain the sequence $y(mT_o)$. The unique representation of a continuous-time signal by its discrete-time representation determined by the Nyquist criterion (3.19) forms the conceptual basis for this process.

The signal flow of the resampling process is shown in Figure 3.9. Frequency domain representations of the signals involved in this process are depicted in Figure 3.10.

The discrete-time Fourier transform $X(e^{j\omega})$ of the sequence $x(nT_i)$ is shown in Figure 3.10a. The D/C conversion operation transforms $x(nT_i)$ into a pulse-modulated continuous-time signal $x_s(t)$. The associated spectrum $X_s(j\Omega)$ is formed by periodic replications of $X(e^{j\omega})$ shifted by multiples of Ω_i as shown in Figure 3.10b. The spectral replications apart from the baseband of the signal in the interval $(-\frac{\Omega_i}{2}, \frac{\Omega_i}{2})$ are termed signal images. To gain the reconstructed continuous-time signal $x_c(t)$, these images have to be removed using a continuous-time reconstruction or anti-imaging filter $h_r(t)$. The ideal reconstruction filter is a lowpass with cutoff frequency $\frac{\Omega_i}{2}$ (3.25). Realizable anti-imaging filters generally introduce errors in the passband of the signals as well as incompletely attenuated images. The latter errors are commonly termed *image components* (see, e.g., [HF00]). They are depicted in Figure 3.10c.

In order to prevent aliasing in the subsequent sampling operation, the signal has to be bandlimited to fulfill the Nyquist criterion (3.19) with respect to the output sampling frequency Ω_o. Therefore, a continuous-time anti-aliasing lowpass filter $h_{aa}(t)$ is utilized to yield the bandlimited signal $x_a(t)$. The corresponding spectrum $X_a(j\Omega)$ is shown in Figure 3.10d. Due to the nonideal characteristics of realizable filters, this operation also introduces passband errors as well as an incomplete suppression of frequencies above the cutoff frequency $\frac{\Omega_o}{2}$ which are termed *aliasing components* in analogy to the reconstruction of continuous-time signals. Finally, the continuous-to-discrete-time conversion C/D transforms $x_a(t)$ into a discrete-time sequence by

(a) Discrete-time input sequence

(b) Pulse-modulated input signal

(c) Reconstructed continuous-time signal

(d) Bandlimited continuous-time signal

(e) Discrete-time output sequence
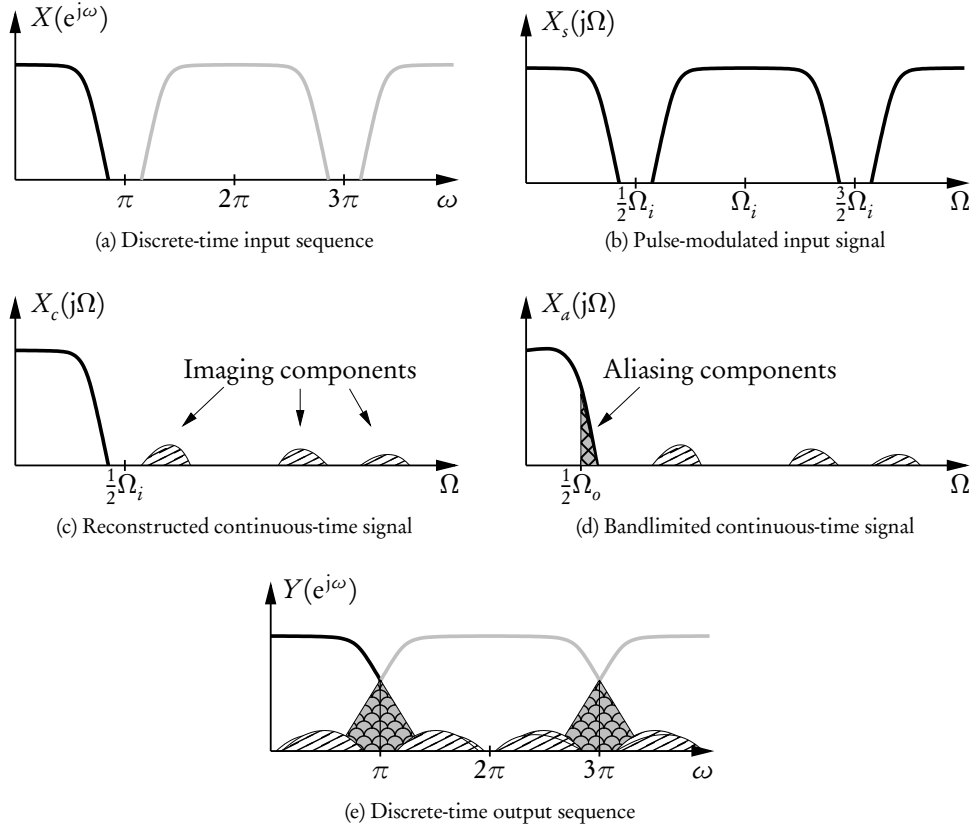
Figure 3.10: Frequency-domain representations of the signals involved in an analog resampling process.

mapping all spectral replications of $X_a(j\omega)$ shifted by integer multiples of Ω_o into each period of the 2π-periodic spectrum $Y(e^{j\omega})$ as depicted in 3.10e.

Within this work, less emphasis is placed on the passband errors caused by the filters $h_r(t)$ and $h_a(t)$, because it is assumed that the passband response of a filter in a discrete-time processing system can be generally designed to meet a desired tolerance. This argument will be justified by the evaluations in Chapter 6 which show that the allowed passband tolerance has relatively little impact on the overall quality and computational complexity of an ASRC algorithm.

Implications from the analog resampling model It can be seen that there are two principal error sources in a sample rate conversion modeled by an analog resampling process: aliasing and imaging. The causes and effects of these errors are explained by the respective models for sampling and reconstruction of continuous-time signals described in Section 3.2. However, because SRC results in a discrete-time signal, both aliasing and imaging components are mapped, i.e. aliased, into the output signal. Nonetheless, it is important to distinguish these errors.

Aliasing due to incompletely suppressed signal images This error occurs irrespective of the conversion rate. It will be referred to as *imaging* in the following.

Aliasing due to imperfect bandwidth limitation This class of errors represents the mapping of components of the baseband of the input signal into the output sequence. It is denoted *baseband aliasing* or simply *aliasing* in the following. Aliasing errors may occur only in case of sample rate reductions, that is, conversion ratios $R < 1$. In this case, deficiencies in the anti-aliasing filter $h_{aa}(t)$ result in incompletely attenuated signal components above $\Omega_o/2$, which are aliased into the discrete-time signal $y(mT_o)$. Equivalently, it can be stated that an anti-aliasing filter $h_{aa}(t)$ is required only if the conversion ratio is less than 1.

Because both the reconstruction filter $h_r(t)$ and the anti-aliasing filter $h_{aa}(t)$ are continuous-time lowpass filters, they can be combined into a single continuous-time filter $h_c(t)$. The ideal frequency impulse and responses of this filter are given by

$$\widehat{h}_c(t) = \widehat{h}_r(t) * \widehat{h}_{aa}(t) \tag{3.53a}$$

$$\widehat{H}_c(j\Omega) = \widehat{H}_r(j\Omega)\widehat{H}_{aa}(j\Omega)$$

$$= \begin{cases} T_i, & |\Omega| < \Omega_c \\ 0, & |\Omega| \geq \Omega_c \end{cases} \quad \text{with } \Omega_c = \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right). \tag{3.53b}$$

In this way, the cutoff frequency Ω_c of $h_c(t)$ depends on the conversion ratio $R = \Omega_o/\Omega_i$ (3.52). In case of an increase of the sample rate, the cutoff frequency is determined by the input sampling rate Ω_i, while for sample rate reductions, the filter response is controlled by the output frequency Ω_o

$$\Omega_c = \begin{cases} \frac{\Omega_i}{2}, & R < 1 \\ \frac{\Omega_o}{2}, & R \leq 1 \end{cases}. \tag{3.54}$$

As a consequence, sample rate conversion is a qualitatively different process depending on whether the sample rate is increased or decreased.

3.4.2 Rational Sample Rate Conversion

Rational sample rate conversion is applicable if the resampling ratio R is expressible by a ratio of integers L and M

$$R = \frac{\Omega_o}{\Omega_i} = \frac{L}{M} \tag{3.55}$$

The motivation to review rational SRC within this work is threefold. First, most algorithms for ASRC evolved from rational SRC, and ASRC is often considered as an extension of rational sample rate conversion. Second, both approaches perform discrete-time signal processing to alter the sampling rate of a signal, and the principal operations and errors in these processes are related. Finally, elements of rational SRC often serve as building blocks for ASRC algorithms.

Rational sample rate conversion is an established and widely-used field of digital signal processing. It is often also referred to as *multirate signal processing*. For more information on this area,
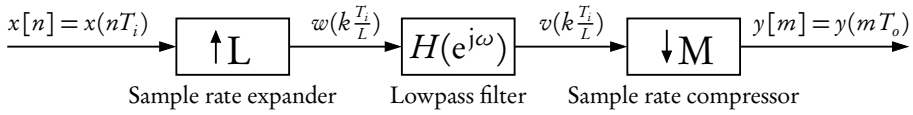
Figure 3.11: Signal flow diagram of a rational sample rate converter.

(a) Block representation

(b) Time-domain behavior

(c) Input and output spectra
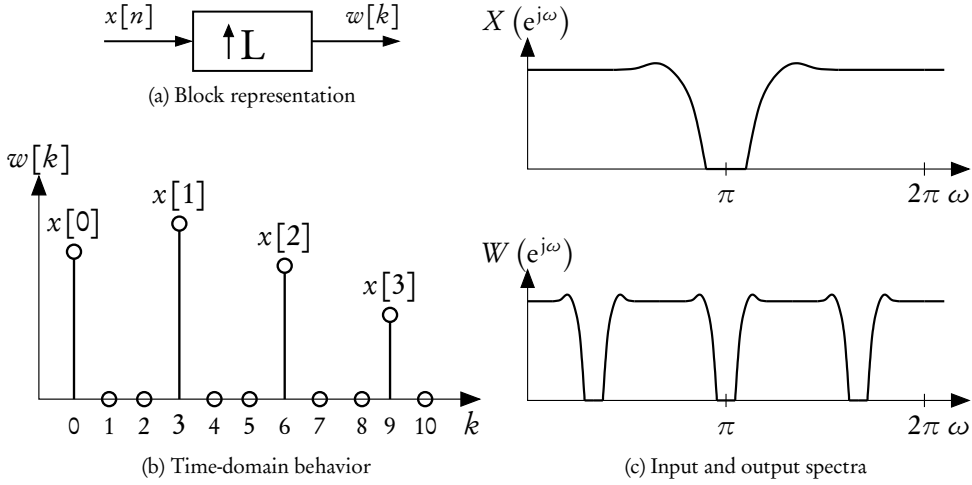
Figure 3.12: Sample rate expander.

see for instance [BBC76, CR81, CR83, Vai92]. Systems for integer-ratio sample rate increases or decreases can be considered as special cases of rational sample rate conversion by setting $M = 1$ or $L = 1$, respectively.

A primary advantage of rational sample rate conversion is that it can be modeled purely by discrete-time linear, periodically time variant signals and systems. A system for implementing a rational sample rate change is depicted in Figure 3.11, consisting of a sample rate expander, a discrete-time lowpass filter $H\left(e^{j\omega}\right)$ and a sample rate compressor.

A sample rate expander is represented in block diagrams by the symbol shown in Figure 3.12a. It increases the sample rate of the signal by an integer factor L by inserting $L-1$ zeros between successive samples of $x[n]$. Formally, this behavior is defined by the linear time-variant operation

$$w[k] = \begin{cases} x[\frac{k}{L}], & k = 0, \pm L, \pm 2L, \dots \\ 0, & \text{otherwise} \end{cases}. \tag{3.56}$$

The resulting sequence is shown in Figure 3.12b. In the frequency domain, this operation corresponds to a compression of the frequency variable by a factor of L. Consequently, each period of the 2π-periodic discrete spectrum $W(e^{j\omega})$ contains L replications of the discrete-time

(a) Block representation

(b) Time-domain behavior

(c) Input and output spectra
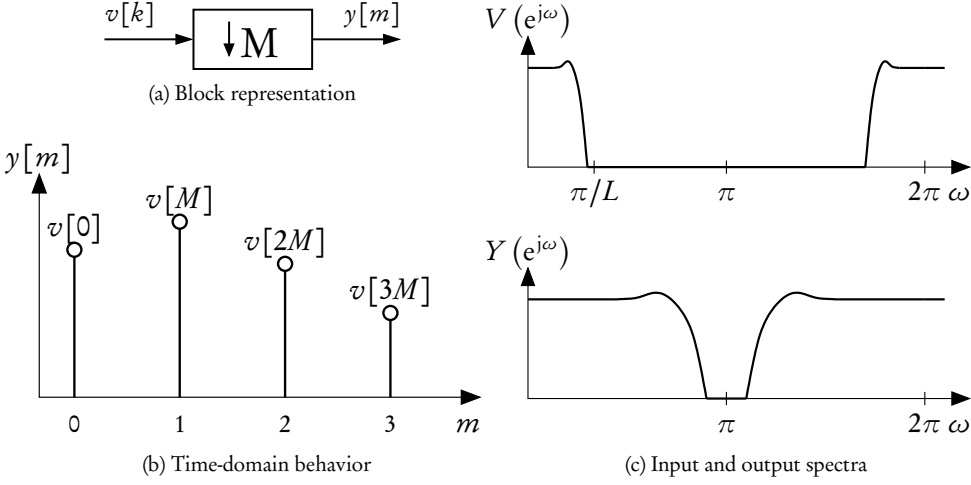
Figure 3.13: Sample rate compressor.

spectrum $X(e^{j\omega})$

$$W\left(e^{j\omega}\right) = X\left(e^{j\omega L}\right). \tag{3.57}$$

The input and output spectra are illustrated in Figure 3.12c. The $L-1$ replications or images of the original spectrum have to be removed by a subsequent discrete-time filter.

A sample rate compressor, represented by the schematic shown in Figure 3.13a, reduces the sampling rate by a factor M by discarding $M-1$ samples out of M values of the input signal $v[k]$ as depicted in Figure 3.13b

$$y[m] = v[Mm]. \tag{3.58}$$

This corresponds to an expansion of the discrete-time spectrum of $V(e^{j\omega})$ by a factor of M. Thus, M shifted replicas of $V(e^{j\omega})$ contribute to each period of the discrete spectrum $Y(e^{j\omega})$.

$$Y\left(e^{j\omega}\right) = \frac{1}{M} \sum_{m=0}^{M-1} V\left(e^{j\frac{\omega-2\pi m}{M}}\right) \tag{3.59}$$

To avoid overlap, that is aliasing, between these shifted replicas, the spectrum $V(e^{j\omega})$ must be bandlimited such that subsequent copies do not overlap. This case is illustrated in Figure 3.13c.

According to this characterization, the discrete-time lowpass filter $H\left(e^{j\omega}\right)$ has two distinct purposes: Removal of the signal images associated with the sample rate expansion and a limitation of the bandwidth prior to sample rate reduction to avoid aliasing. While both aspects can be implemented in a single filter, the determination of the cutoff frequency of the filter depends on the conversion ratio. The ideal filter specification is given by [CR83]

$$\widehat{H}\left(e^{j\omega}\right) = \begin{cases} L, & |\omega| < \min\left(\frac{\pi}{L}, \frac{\pi}{M}\right) \\ 0, & \text{otherwise} \end{cases}. \tag{3.60}$$

This filter specification is in accordance with the design specification of the combined continuous-time anti-imaging/anti-aliasing filter of the SRC model based on analog resampling (3.53b), although $H(e^{j\omega})$ is a purely discrete-time filter. In case of an overall increase of the sample rate (conversion ratio $R > 1$), the cutoff frequency is determined by the upsampling ratio L, while for an overall sample rate reduction ($R < 1$), the filter specification is controlled by M.

It is noted that the passband gain of L in (3.60) maintains the frequency normalization inherent to a sampled signal representation (3.20) after the sampling rate is changed. However, the sample rate decrease by M does not require such an adjustment. As observed in (3.59), the required scaling is integrated into the operation of the sample rate compressor [CR83, OSB99].

In summary, while rational SRC is based on purely discrete-time algorithms, the basic sources of errors, aliasing and imaging, are identical to the processes of sampling and reconstruction as well as to ASRC problems.

3.4.2.1 Polyphase Implementations for Rational SRC

As shown in the previous section, rational SRC is defined in terms of sample rate expanders, compressors, and digital filters. However, for efficient implementations, sophisticated structures such as polyphase networks or commutator models are used instead of these building blocks. We introduce some basic concepts of polyphase filtering here for two reasons: First, integer- or rational-ratio SRC is part of numerous FD or ASRC algorithms. Second, several concepts of ASRC are best described and understood in terms of polyphase filtering, even if the implementation does not contain rational SRC. Only a brief characterization of polyphase techniques is provided here. For a more comprehensive treatment, the reader is referred to the cited references on multirate signal processing or [Hen02].

The basic idea for an efficient implementation of a 1-to-L sample rate increase is that the filtering operation by $H(e^{j\omega})$ includes a large fraction of multiplications by zero due to the zeros inserted into the input signal of the filter by the sample rate expander. Therefore, only a small number of impulse response coefficients $h[n]$ of $H(e^{j\omega})$ are actually used in the computation of a particular output value $y[m]$. For this reason, the impulse response $h[n]$, which is commonly referred to as the *prototype filter* of the SRC system, can be partitioned into subsets that contribute to an output sample $y[m]$. As rational SRC generally forms linear periodically time-variant systems, these subsets repeat periodically. Within this work, only FIR filters are considered for $H(e^{j\omega})$ due to their beneficial properties for this application (see, e.g. [SR73, CR83]). In this case, there are L finite subsets which are referred to as *polyphase filters* or *polyphase branches*

$$h_l[n] = h[nL + l] \qquad \text{for } l = 0, 1, \ldots, L-1; \quad n = 0, 1, \ldots, Q-1. \tag{3.61}$$

Here, Q denotes the length of the polyphase filters. For notational convenience, it is assumed that the length of the prototype filter, $N + 1$, is an integer multiple of the upsampling ratio L. This can be ensured without loss of generality by zero-padding $h[n]$ appropriately. Thus, Q is determined by

$$Q = \frac{N+1}{L}. \tag{3.62}$$

Utilizing the polyphase representation, the calculation of an output sample can be formulated as

(a) 1-to-L sample rate increase

(b) M-to-1 sample rate decrease
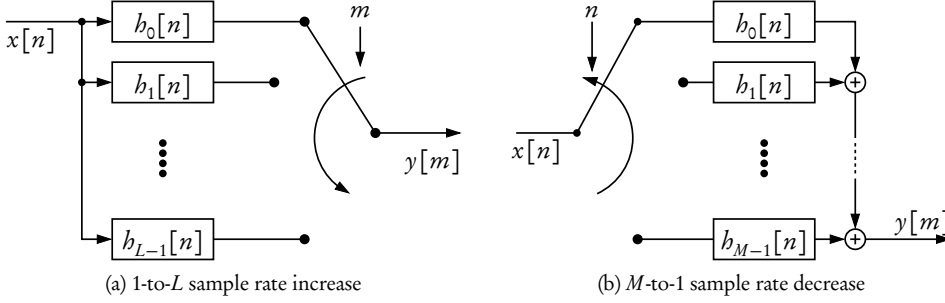
Figure 3.14: Commutator structures for integer-ratio sample rate converters.

a convolution

$$y[m] = \sum_{k=0}^{Q-1} h_{m \bmod L}[k] x \left[\left\lfloor \frac{m}{L} \right\rfloor - k \right] , \tag{3.63}$$

where mod denotes the modulo operation [GKP06]

$$z = x \bmod y \quad \Longleftrightarrow \quad z = n \cdot x + y \text{ and } n \in \mathbb{Z}. \tag{3.64}$$

This operation is conveniently modeled or implemented using a commutator model as shown in Figure 3.14a. The input signal $x[n]$ is filtered in parallel by the L polyphase branches. The rotating commutator selects one of these outputs for each output sample $y[m]$. In this way, the polyphase branches operate at the low input frequency Ω_i.

For an integer-ratio sample rate decrease, the observation that $M-1$ out of M filtered values $v[k]$ are discarded by the compressor can be utilized for efficient implementation. In terms of filtering, for each sample of the intermediate signal $v[k]$, only a fraction of the products in the discrete convolution $v[k] = w[k] * h[n]$ contributes to the final output $y[m]$. As in case of integer sample rate increases, the prototype filter can be partitioned into a set of polyphase filters $h_m[n]$

$$h_m[n] = h[nM + m] \quad \text{for } m = 0, 1, \ldots, M-1; n = 0, 1, \ldots, Q-1 \text{ with } Q = \frac{N+1}{M}, \tag{3.65}$$

assuming that the prototype filter length is an integral multiple of M. Thus, the computation of an output sample results is performed as

$$y[m] = \sum_{l=0}^{M-1} \sum_{k=0}^{Q-1} h_l[k] x \left[(m-k)M + l \right] . \tag{3.66}$$

Again, this operation is conveniently represented by a commutator model as shown in Figure 3.14b. The rotating commutator distributes the input samples to the M polyphase branches that consequently operate at the low output sampling frequency $\Omega_o = \Omega_i / M$. The output samples $y[m]$ are formed by summing the outputs of the polyphase branches.

3.4.2.2 Use of the Term Interpolation

The term "interpolation" is used ubiquitously in DSP, in particular in the fields of FD filtering and sample rate conversion, often with indistinct or contradicting meanings. In this way, it is a common source of misconceptions.

In the fields of approximation theory and numerical analysis, interpolation refers to the construction of a function $f(x)$ from a set of data points $\{(x_i, y_i)\}$, $i = 0, 1, \ldots, N$ that coincides exactly with the function values at the abscissas of these points [CL00, Pow91, PTVF92, Mei02]

$$f(x_i) = y_i \qquad \text{for } i = 0, 1, \ldots, N. \tag{3.67}$$

Equation (3.67) is commonly referred to as the interpolation condition, which has been stated in (3.33) in the context of sampling and reconstruction.

In this way, interpolation is clearly distinct from approximation, which is also defined as the construction of a function from a set of function values, but does not require the approximating function to pass through these points. In [Sch46a], interpolation and approximation are denoted as ordinary and smoothing interpolation formulae, respectively.

In rational SRC, the term interpolation is used for a sample rate change where the sampling frequency is increased, e.g. $f_o > f_i$ [CR81, CR83]. Likewise, a reduction of the sample rate is termed *decimation*. Although the interpolation condition (3.67) is noted as a possible condition in the filter design specification for sample rate increases, it is not considered as an obligatory requirement [CR83]. In the literature on FD filtering and ASRC, the term interpolation often does not imply the interpolation condition (e.g. [EGH93, LVKL96, Ves99, VS07]). Therefore, the described operation would be better characterized as approximation.

On the other hand, the adherence to the interpolation condition is not mandatory for a wide range of signal processing applications, see e.g. [EGH93]. Instead of forcing the error to zero on a finite set of points, an approximation error that is minimal with respect to a given norm over the complete approximation interval is more desirable for most applications.

For these reasons, we restrict the use of the term interpolation to systems that conform to the interpolation condition. Digital filters that approximate the signal between samples of a discrete-time sequence will be termed *resampling filter* in the following. Likewise, in the context of integer-ratio or rational SRC, the term "sample rate increase" will be used instead of interpolation.

3.4.2.3 Limitations of Rational Sample Rate Conversion

Despite its many uses, rational SRC shows a number of shortcomings that reduce its applicability for various problems:

- If the numerator and denominator terms of the conversion ratio $R = \frac{L}{M}$ are large coprime integers, then rational SRC becomes impractical. Such conversion ratios are often denoted as *incommensurate*, e.g. in [CR83, LVKL96]. Although the computational effort does not necessarily increase in such cases, memory requirements, control effort and the requirements on the prototype filter design often become limiting factors. In literature, the distinction is often drawn between rational and irrational conversion ratios, for instance in [Ram84, Gar93, SR96, Eva00b, Hen02]. However, in the author's opinion, this classification is moot, because for implementations of SRC on digital hardware, conversion

ratios are invariably expressed as rational numbers. However, rational SRC is appropriate only if the conversion ratio R is representable as a ratio of integer numbers L and M of moderate size.

- Because the conversion ratio R influences the design of the discrete-time filter $H(e^{j\omega})$, a rational SRC system can be used only for a single, fixed ratio. Therefore, processing systems utilizing different conversion ratios need additional memory for multiple coefficient sets, and possibly require control logic to switch between different conversion ratios.

- Interfacing between asynchronous systems, which are not synchronized to a common clock, results in time-varying conversion ratios that may assume arbitrary values within a given interval. Due to the restriction to a single ratio or a fixed set of conversion ratios, rational SRC is not suited for this class of applications.

3.4.3 Advantages and Applications of Arbitrary Sample Rate Conversion

Arbitrary sample rate conversion aims at overcoming most deficiencies of rational SRC. First, it enables arbitrary conversion ratios and is not limited to ratios expressible by moderately sized values L and M. Moreover, the effort in terms of memory requirements, control effort, and computational costs is largely independent of the precise conversion ratio. Second, as the design of ASRC systems is irrespective of a particular conversion rate, variable ratios are supported. This can be used either to reduce memory requirements for applications that utilize a fixed set of conversion ratios, or to enable continuously varying ratios, for instance for interfacing asynchronous systems, e.g. [LK81b, Gar93].

It is important to note that the independence of the conversion ratio typically attributed to ASRC is limited by one fundamental distinction. As argued in Section 3.4.1 and manifested in the ideal anti-aliasing/anti-imaging filter $\widehat{H}_c(j\Omega)$ (3.53), arbitrary sample rate conversion is a qualitatively different process depending on whether the conversion ratio R is larger or smaller than unity. While for sample rate increases, the characteristics of $\widehat{H}_c(j\Omega)$ are constant, they depend on R in case of sample rate decreases.

Seemingly for this reason, ASRC algorithms are often limited to sample rate increases and are, for instance, termed interpolation filters (e.g. [VS96a, VS07]). Here, "interpolation" has to be interpreted in the sense used in multirate signal processing (see Section 3.4.2.2).

In general, ASRC systems that facilitate increasing and decreasing sampling rates either utilize different implementation structures [HF00, Hen02, BVSR02, KG08], or switch between different operation modes, e.g. [LPW82, SG84].

Within this work, emphasis is placed on ASRC systems for sample rate increases for two reasons. First, as will be shown in Chapter 4, only this class of algorithms is suitable for efficient signal processing techniques for WFS. Second, structures and filter designs for sample rate decreases can be typically obtained by transforming systems for increasing the sampling rate.

Applications of arbitrary sample rate conversion include:

Timing recovery in digital modems   [Gar93, EGH93, VS96a] In a digital modem, the symbols of the incoming signal must be synchronized to the clock of the digital processing system. Thus, the ability to interface asynchronous systems is the decisive advantage of ASRC in this application.

Software defined radio [HF00, Hen02, AS06] To support a growing number of mobile communication standards, functionality such as demodulation, which is traditionally implemented in the analog domain, is shifted toward software realization in the discrete-time domain. Here, sample rate conversion forms a major part of the computational effort. ASRC methods are preferred over rational or integer SRC algorithms due to complexity and memory requirements. Moreover, ASRC is capable of handling different conversion ratios to support different carrier frequencies without additional memory requirements.

Sample rate conversion for digital audio [LK81a, LPW82, SG84, CDPS91, AK93, RLF00] In digital audio, a multitude of digital audio sampling frequencies, for instance 48 or 96 KHz for professional applications and 44.1 KHz for consumer electronics, is in use. Therefore, high-quality sample rate conversion techniques are required both for mastering digital media and for realtime transmission. In many cases, the numerator and denominator of the conversion ratio are relatively large coprime numbers. For example, the rational conversion ratio for a transformation from 48 KHz to 44.1 KHz is $L/M = 147/160$. Consequently, a direct application of rational SRC requires long prototype filters and either a very high intermediate frequency or a sophisticated multistage implementation at the cost of a complex control logic. Using ASRC, both control effort and coefficient storage can be reduced significantly. Moreover, many audio applications based on ASRC techniques also provide synchronization between asynchronous interfaces.

3.4.4 Models for Asynchronous Sample Rate Conversion

Modeling the behavior of an ASRC system is necessary to assess the performance of an algorithm as well as for purposefully designing such algorithms. However, modeling ASRC algorithms is more complex than the characterization of purely discrete-time, single-rate systems, such as digital filters or variable fractional delay structures, for two primary reasons. First, the coexistence of two different sampling rates Ω_i and Ω_o inhibits the use of conventional models for linear time-invariant discrete-time systems. The ability of ASRC systems to handle variable conversion ratios introduces an additional degree of freedom, putting further demands on the model. Two basic approaches, either based on discrete-time or continuous-time models, are described in the following.

3.4.4.1 Discrete-Time Models

In multirate signal processing, bi-frequency system functions (also denoted as bifrequency maps or bispectrum maps) [Zad50, CR81, AV00, Hen02] are an established tool to describe linear periodically time-variant systems such as rational sample rate converters. The bi-frequency system function $K(e^{j\omega'}, e^{j\omega})$ defines a mapping from a periodic input spectrum with normalized angular frequency ω to a periodic output spectrum with frequency variable ω' corresponding to a potentially different sampling frequency

$$K\left(e^{j\omega'}, e^{j\omega}\right) = \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} k[m,n] e^{-j\omega' m} e^{j\omega n}. \tag{3.68}$$

This system function is not restricted to one-to-one mappings between input and output frequencies, but may also represent one-to-many, many-to-one, or many-to-many mappings between

frequencies. In discrete-time sample rate conversion, such sophisticated mappings result from imaging or aliasing artifacts. The function $k[m,n]$ in (3.68) denotes the discrete-time Green's function of the system,[1] which describes the response of the system at time m for a unit impulse applied at input sampling time n

$$y[m] = \sum_{n=-\infty}^{\infty} k[m,n]x[n].$$ (3.69)

Thus, $k[m,n]$ is distinct from the time-varying impulse response $h[m,n]$ corresponding to the convolution

$$y[m] = \sum_{n=-\infty}^{\infty} h[m,n]x[m-n],$$ (3.70)

but are related by

$$h[m,n] = k[m,m-n].$$ (3.71)

In contrast to the impulse response $h[m,n]$, the discrete-time Green's function is also applicable if the input and output sampling rates of the systems are different.

Using the bi-frequency system function (3.68), the output spectrum of the system is determined by

$$Y\left(e^{j\omega'}\right) = \int_{-\pi}^{\pi} K\left(e^{j\omega'}, e^{j\omega}\right) X\left(e^{j\omega}\right) d\omega.$$ (3.72)

However, this system transmission function is a practical tool only if the Green's function $k[m,n]$ or, equivalently, the time-varying impulse response $h[m,n]$ are at least periodically time-variant functions. For sample rate conversion systems, this condition is equivalent to a rational conversion ratio R. In this case, each input frequency ω maps only to a finite set of output frequencies ω'. An example for a bi-frequency mapping for a rational conversion ratio $R = 4/5$ is shown in Figure 3.15.

Discrete-time approaches for modeling ASRC algorithms, although limited to rational conversion ratios, have been proposed in [BLR03], [LR03], [LBR04]. In [LR03], an alternative to bi-frequency maps based on the combination of sampled continuous-time impulse responses is presented. [BLR03] considers discrete-time models for polynomial-based ASRC algorithms for fixed, rational conversion ratios, and derives its relation to conventional multirate structures. This analogy is also discussed in [Hen02]. Discrete-time models for ASRC algorithms utilizing a large upsampling ratio L to create a dense grid of intermediate samples are compared to continuous-time models in [LBR04], concluding that discrete-time models can be used to approximate continuous-time models in most applications of SRC.

In summary, the restrictions to fixed and rational conversion ratios are the most severe limitations of discrete-time models for ASRC algorithms.

---

[1]Conceptually, the discrete-time Green's function is closely related to the Green's function $G\left(\mathbf{x}|\mathbf{x}_s, \omega\right)$ for acoustic wave propagation used in Section 2.1. While the latter describes the response at position $\mathbf{x}$ caused by an elementary acoustic source with frequency ω at position $\mathbf{x}_s$, $k[m,n]$ models the system response at time index m to an unit impulse, that is, an elementary excitation, at time n.
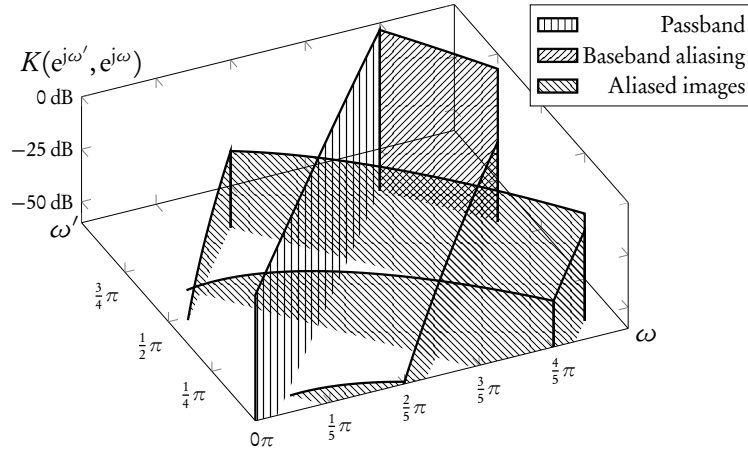
Figure 3.15: Bi-frequency system response for rational sample rate conversion with ratio $R = 4/5$. ASRC system based on Lagrange interpolation, order $N = 3$.
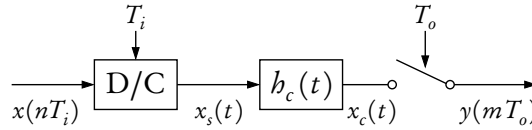
Figure 3.16: Hybrid analog/digital model for arbitrary sample rate conversion.

3.4.5 Continuous-Time Description: The Hybrid Analog/Digital Model

Unlike discrete-time descriptions, continuous-time models are based on the analog resampling interpretation of ASRC described in Section 3.4.1. This interpretation is widely used in ASRC and is commonly referred to as the *hybrid analog/digital model* [Gar93, VS96b, Ves99, GBLR04, VS07]. A block diagram of this model is shown in Figure 3.16.

An ideal D/C converter transforms the discrete-time input sequence $x(nT_i)$ into a pulse train $x_s(t)$ according to (3.4c). The continuous-time filter $h_c(t)$ serves as a combined anti-aliasing/anti-imaging filter according to (3.53). Subsequent sampling with period T_o yields the discrete-time output sequence $y(mT_o)$.

The most important advantage of the hybrid analog/digital model is that the behavior of the ASRC system is completely determined by the continuous-time impulse response $h_c(t)$ or, equivalently, its continuous-time frequency response $H_c(j\Omega)$

$$H_c(j\Omega) = \mathscr{F}\{h_c(t)\}. \tag{3.73}$$

In Figure 3.17, the continuous-time impulse and frequency responses of a typical ASRC system are shown. The continuous-time filter $h_c(t)$ approximates the ideal filter characteristic of an ana-

(a) Impulse response $h_c(t)$

(b) Zero-phase frequency response $H_c(j\Omega)$
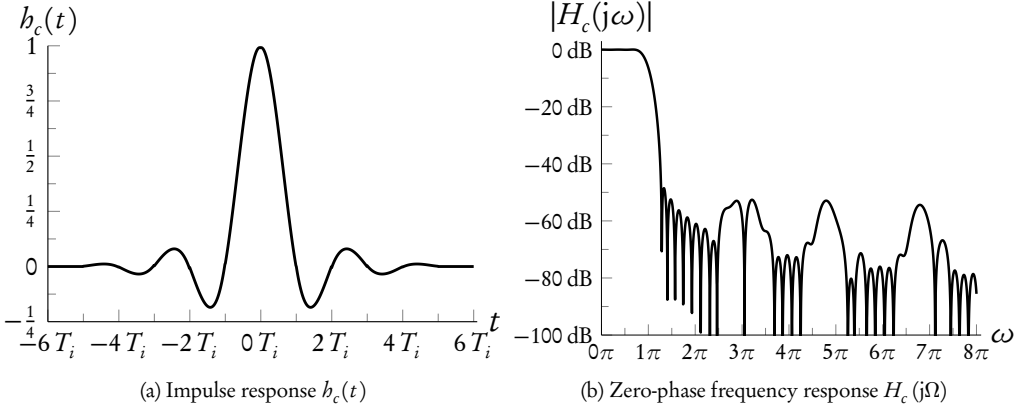
Figure 3.17: Continuous-time impulse and frequency responses of an ASRC system.

log resampler (3.53), that is, an ideal lowpass filter with cutoff frequency $\Omega_c = \min(\Omega_i/2, \Omega_o/2)$

$$\widehat{H}_c(j\Omega) = \begin{cases} T_i, & |\Omega| < \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \\ 0, & |\Omega| \geq \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \end{cases}.$$

where the passband magnitude of T_i results from the characteristics of the ideal reconstruction filter (3.25).

In this way, the principal sources of errors for sample rate conversion stated in Section 3.4.1, namely aliasing, imaging and passband deviations, can be assessed directly by comparison to the ideal frequency response.

As a second advantage, the hybrid analog/digital model is applicable to most ASRC methods. In particular, this applies to methods that can be represented by a continuous-time impulse response $h_c(t)$, which holds for the vast majority of existing algorithms.

Finally, analysis using the hybrid analog/digital model is independent of the conversion ratio R. Examination of the continuous frequency response enables an evaluation of passband errors as well as aliasing and imaging components without evaluating the frequency response for a particular R.

However, the utility of the hybrid analog/digital model is affected by the qualitative distinction between increasing and decreasing conversion ratio explained in Section 3.4.3. For nondecreasing conversion ratios, that is $\Omega_o \geq \Omega_i$, the characteristics of $\widehat{H}_c(j\omega)$ are constant. Thus, the performance can be analyzed irrespective of a particular conversion ratio R. In contrast, for decreasing ratios, the cutoff frequency of $\widehat{H}_c(j\omega)$ is a function of R. Consequently, the conversion ratio has to be accounted for in the analysis. Nonetheless, this is not a limitation of the hybrid analog/digital model, but results from intrinsic properties of SRC systems.

As stated above, this work largely focuses on algorithms for nondecreasing sampling rates. Therefore, a single continuous-time impulse or frequency response is appropriate to characterize an ASRC algorithm irrespective of a particular conversion ratio.

It is remarkable that the idea of characterizing a practical interpolation or approximation algorithm by means of a single, continuous function of compact support predates digital signal processing significantly. To the author's knowledge, this concept was first used by I.J. Schoenberg in the landmark series of papers that introduced spline functions [Sch46a]. For a historical view, see, for instance, [Mei02]. In turn, Schoenberg refers to the cardinal interpolation formula (3.31b) introduced in [Whi15, Whi29] as an example of an interpolation method based on a continuous function, but deems it inappropriate for practical purposes due to its slow damping. Moreover, Schoenberg introduces the use of the Fourier transform of this defining function, which is referred to as the characteristic function of the interpolation formula, and expresses properties of the interpolation method in terms of this Fourier transform.

Despite the continuous-time description, ASRC algorithms are generally implemented on digital architectures by means of discrete-time signal processing. For this purpose, the continuous representation is evaluated only at a finite set of points that are determined by the output sampling time. This approach is equivalent to sampling the continuous-time impulse response $h_c(t)$. The sampled values are utilized in a discrete-time filtering process. In the vast majority of methods, the sampled impulse response contains only a finite number of nonzero values. Therefore, on the most basic level, most ASRC algorithms are based on finite discrete convolutions.

3.5 Arbitrary Delay Operations as Discrete-Time Filtering Processes

As pointed out in the preceding sections, the basic operation of both VFD filtering and ASRC is the determination of signal values at arbitrary positions between the samples of a given discrete-time sequence. Moreover, we have shown that for both applications, the signal processing operation is best implemented as a discrete-time convolution of finite length.

In order to describe different algorithms for VFD filtering and ASRC in a uniform way, a common notation is introduced in this section. The notion of basepoint sets enables a coherent description of different algorithms and different indexing schemes used in literature. In addition, common operations such as the determination of the intersample position and the input samples to be used for resampling can be represented consistently.

A central difference between FD and ASRC is the orientation of variables to denote the positions of output samples. In FD filtering, a fractional delay value d, which has a reverse orientation with respect to the time variable t, is utilized. In ASRC applications, the intersample position μ denotes the location of an output value in relation to the input samples $x[n]$. The quantity μ has the same orientation as the time axis. To enable a common notation, the intersample position μ is used consistently throughout this work. However, this choice does not limit the generality of the representation. Section 3.5.2 describes transformations between the notations for FD filtering and ASRC.

3.5.1 Uniform Notation Using Basepoint Sets

In its most general form, resampling by means of a finite discrete-time convolution is represented by [EGH93, VS07]

$$y(t) = \sum_{k=N_1}^{N_2} x[n-k]h(k,\mu). \tag{3.74}$$

The index limits N_1 and N_2 affect the samples $x[n]$ involved in the discrete convolution. Moreover, they determine the order of the resampling filter $N = N_2 - N_1$.

The integer-valued sample index n and the intersample position μ are functions of the output time t. A common choice for these terms is

$$t = (n+\mu)T_i \qquad \text{with } n \in \mathbb{N}. \tag{3.75}$$

This calculation normalizes the intersample position μ with respect to the input sampling period $T_i = 1/f_i$. Thus, the filter coefficients $h(k,\mu)$ are independent of the sampling frequency.

The quality of resampling depends on the location of the intersample position with respect to the samples used in the convolution. In general, n has to be chosen such that the output instant t lies in the central interval of the involved samples [EGH93]. As a consequence, the choices for N_1, N_2 and the calculation of n and μ are tightly interrelated. For each choice, there is a unit-length interval for μ that corresponds to the region of optimal resampling quality

$$\mu_{min} \leq \mu \leq \mu_{max} \qquad \text{with } \mu_{max} - \mu_{min} = 1. \tag{3.76}$$

As argued in Section 3.3.3 for the case of FD filtering, a closed interval is used here without loss of generality, while resolving the ambiguity at the end points of the interval is left to the actual implementation.

Due to the degrees of freedom in the specification of N_1, N_2, n and μ, several specialized forms of the convolution (3.74) are in use. In

$$y(t) = \sum_{k=0}^{N} x[n-k]h(k,\mu), \tag{3.77}$$

the indices are natural numbers $k = 0, 1, \ldots, N$. In this case, n and μ are given by

$$n = \text{round}\left(\frac{t}{T_i} + \frac{N}{2}\right) \tag{3.78a}$$

$$\mu = \frac{t}{T_i} - n. \tag{3.78b}$$

Thus, μ falls within the optimal range

$$-\frac{N+1}{2} \leq \mu \leq -\frac{N-1}{2} \tag{3.79}$$

This form is widely used for FD filtering [LVKL96, Väl95a]. The corresponding interval for the fractional delay part d is

$$\frac{N-1}{2} \leq d \leq \frac{N+1}{2}. \tag{3.80}$$

Another common form of (3.74) is used, among others, in [VS96a, VS07]

$$y(t) = \sum_{k=-\frac{N+1}{2}}^{\frac{N-1}{2}} x[n-k]h(k,\mu) \qquad \text{with} \tag{3.81a}$$

$$n = \left\lfloor \frac{t}{T_i} \right\rfloor \tag{3.81b}$$

$$\mu = \frac{t}{T_i} - n. \tag{3.81c}$$

Consequently, the intersample position interval is given by

$$0 \leq \mu \leq 1. \tag{3.81d}$$

Note that the limits in the summation (3.81) differ from the original references, since N denotes the filter length there, as opposed to the filter order N used here consistently. In any case, this formulation is restricted to odd orders N, because the sample indices of $x[n-k]$ would not denote integer numbers for N even.

Such restrictions as well as the different notations in use complicate the analysis and the comparison of algorithms that are expressed in different forms. For this reason, a unified description based on the notation of a *basepoint set* is emphasized here. The term basepoint set is chosen in the style of [EGH93]. This notation has been introduced and has been elaborately applied by the author in the analysis of Lagrange FD filters in [Fra08].

The basepoints of a resampling algorithm, denoted $\{t_k\}$, is a set of $N+1$ real numbers that denote the time values, i.e. abscissas, of the discrete samples $x[n-k]$ with respect to the intersample position, as shown in Figure 3.18. The basepoints form a decreasing sequence with unit distance

$$\{t_k\} = t_0, t_1, \ldots, t_N \tag{3.82a}$$

$$t_k = t_0 - k \qquad \text{for } 0 \leq k \leq N. \tag{3.82b}$$

So, the basepoints are normalized to the input sampling period T_i. The decreasing order is different to the basepoint set notation used for FD filtering [Fra08], where the basepoints b_k form an increasing sequence. This distinction is due to the converse orientations of the fractional delay value d and the intersample position μ.

Using the notation of basepoint sets, the discrete convolution is stated with the index range $N_1 = 0$ and $N_2 = N$

$$y(t) = \sum_{k=0}^{N} x[n-k]h(k,\mu), \tag{3.83}$$
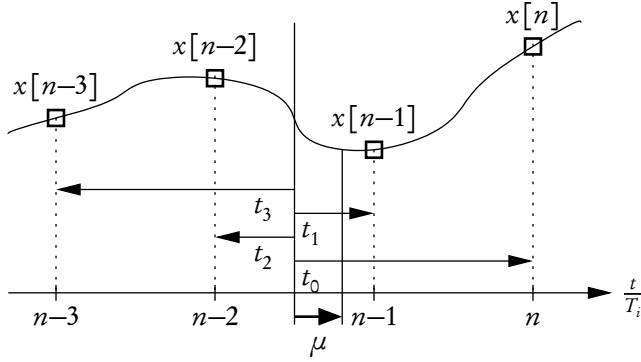
Figure 3.18: Basepoint set notation for resampling based on a finite discrete convolution.

that is, similar to the form (3.77). However, the use of basepoint sets decouples the optimal range of the intersample position (3.76) from the choice of the index range. In this way, the restriction of μ to the central interval of the basepoint set results in

$$\mu_{min} = \frac{t_0 + t_n - 1}{2} \qquad \text{and} \qquad \mu_{max} = \frac{t_0 + t_n + 1}{2} . \tag{3.84}$$

The sample index n is calculated as

$$n = \text{round}\left(\frac{t}{T_i} + \frac{N}{2}\right) . \tag{3.85}$$

As a notable consequence, the determination of the samples involved in the convolution is irrespective of the particular basepoint set in use and depends only on the order N. This follows from the restriction of the intersample position to the central interval of the basepoint set. The intersample position is calculated as

$$\mu = \frac{t}{T_i} + t_0 - \text{round}\left(\frac{t}{T_i} + \frac{N}{2}\right) \tag{3.86a}$$

$$= \frac{t}{T_i} + t_0 - n . \tag{3.86b}$$

In this way, the simple identity $t = (n + \mu)T_i$ (3.75) does no longer hold. Instead, this notation enables arbitrary time offsets associated with the samples $x[n - k]$.

The convolution sum with index variables $k = 0, \ldots, N$ (3.77) is characterized by the basepoint set

$$\{t_k\} = 0, -1, \ldots, -N \tag{3.87a}$$

$$t_k = -k \qquad \text{for } 0 \le k \le N . \tag{3.87b}$$

Likewise, the filtering operation (3.81) is expressed by

$$\{t_k\} = \frac{N+1}{2}, \frac{N-1}{2}, \ldots, -\frac{N-1}{2} \tag{3.88a}$$

$$t_k = \frac{N+1}{2} - k \qquad \text{for } 0 \leq k \leq N. \tag{3.88b}$$

It is notable that the use of basepoints resolves the restriction to odd orders that is inherent to (3.81).

Besides the unified notation for the discrete convolution, the basepoint set formalism is especially useful for algorithms that use the time values associated with the samples $x[n-k]$ to compute the filter coefficients $h(k, \mu)$. Methods originating in numerical interpolation or approximation techniques such as Lagrange or spline interpolation are examples of such algorithms. In these cases, the choice of the basepoint set potentially affects the efficiency of these methods.

Basepoint sets that are symmetric with respect to zero form a particularly important class, which is given by

$$\{t_k\} = \frac{N}{2}, \frac{N}{2} - 1, \ldots, -\frac{N}{2} \tag{3.89a}$$

$$t_k = \frac{N}{2} - k \qquad \text{for } 0 \leq k \leq N, \tag{3.89b}$$

resulting in the symmetry condition

$$t_{N-k} = -t_k \qquad \text{for } 0 \leq k \leq N. \tag{3.89c}$$

Following from (3.84), the intersample position range is also symmetric with respect to zero

$$-\tfrac{1}{2} \leq \mu \leq \tfrac{1}{2}. \tag{3.89d}$$

Symmetric basepoint sets show a number of advantageous properties, for instance coefficient symmetries in the calculation of the discrete filter taps $h(k, \mu)$ that potentially result in more efficient algorithms. The impact of basepoint symmetries is investigated in detail in [Fra08] for the case of Lagrange interpolation. Nonetheless, these principles apply to other classes of resampling filters as well. This will become apparent in Section 3.6.

3.5.2 Application to Fractional Delay Filtering

As stated above, the conversion between the notation based on an ASRC process and the formulation of a FD filter is straightforward.

According to (3.34), the output sequence $y[m, D]$ of a FD filter approximates the reconstructed continuous-time signal $x_c(t)$ delayed by a value of DT_i, where T_i denotes the sampling frequency used in the reconstruction of $x_c(t)$ from a sequence $x[n]$.

$$y[m, D] = x_c\left([m - D]\, T_i\right). \tag{3.90}$$

Substituting $x_c(t)$ with the discrete filtering operation of a resampling algorithm (3.83) yields

$$y[m, D] = \sum_{k=0}^{N} x[n-k] h(k, \mu). \tag{3.91}$$

Applying $t = [m - D]T_i$ to (3.85) and (3.86) results in explicit expressions for n and μ

$$
\begin{aligned}
n &= \text{round}\left(\frac{(m-D)T_i}{T_i} + \frac{N}{2}\right) \\
&= m - \text{round}\left(D - \frac{N}{2}\right) & \text{(3.92a)}
\end{aligned}
$$

$$
\begin{aligned}
\mu &= \frac{(m-D)T_i}{T_i} + t_0 - n \\
&= m - D + t_0 - \left[m - \text{round}\left(D - \frac{N}{2}\right)\right] \\
&= \text{round}\left(D - \frac{N}{2}\right) + t_0 - D. & \text{(3.92b)}
\end{aligned}
$$

In this way, the determination of the intersample position μ and the sample index n for a FD filter is expressed in the formalism of a resampling process for arbitrary basepoint sets.

However, it is possible to transform these relations into a form more related to FD filtering. The difference between the indices n and m represents the integer part D_{int} of the FD element (3.47)

$$
\begin{aligned}
D_{int} &= m - n & \text{(3.93a)} \\
&= \text{round}\left(D - \frac{N}{2}\right). & \text{(3.93b)}
\end{aligned}
$$

Therefore, FD filtering based on a finite discrete convolution can be expressed by

$$
y[m, D] = \sum_{k=0}^{N} x[m - D_{int} - k]h(k, \mu) \tag{3.94a}
$$

$$
= \sum_{k=0}^{N} x[m - D_{int} - k]h(k, d), \tag{3.94b}
$$

where $h(k, d)$ denotes the filter coefficients of a FIR FD filter for a fractional delay value d. To relate $h(k, d)$ to the filter coefficients $h(k, \mu)$, a substitution of (3.47) into (3.86) yields

$$
d = t_0 - \mu \qquad \Longleftrightarrow \qquad \mu = t_0 - d. \tag{3.95}
$$

3.5.3 Representation as Continuous-Time Impulse Response Function

Aside from notational issues such as the choice of the basepoints or the intersample position range, resampling based on a discrete convolution formula (3.83) is mainly characterized by the filter coefficients

$$
h(k, \mu) \qquad \text{for } \mu_{min} \le \mu \le \mu_{max} \text{ and } k = 0, 1, \dots, N.
$$

Alternatively, a resampling filter can be represented by a single continuous-time function, the continuous-time impulse response $h_c(t)$. The benefits of this representation are twofold.
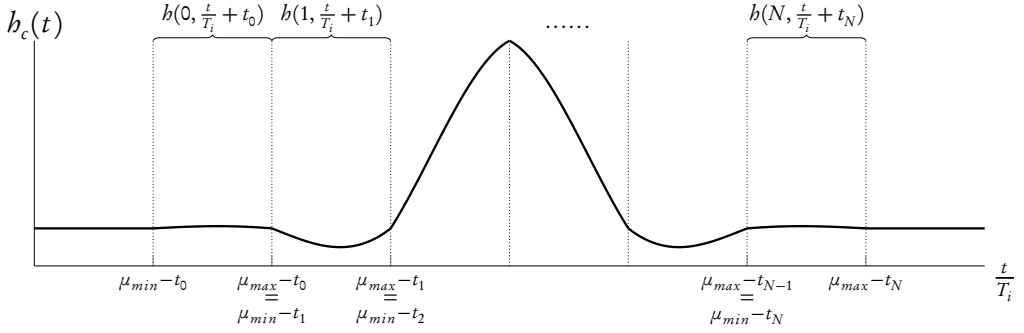
Figure 3.19: Continuous-time impulse response $h_c(t)$ constructed from a resampling function based on a finite discrete-time convolution.

First, the convolution sum (3.83) can be stated compactly as

$$y(t) = \sum_{k=0}^{N} x[n-k] h_c \left([\mu - t_k] T_i \right),$$

(3.96)

where the discrete filter coefficients $h(k, \mu)$ are obtained by sampling $h_c(t)$ [VS96a, HF00, Hen02, VS07].

The form used here builds upon the notation of basepoint sets and is therefore independent of a particular index range or the intersample position interval. The minus sign in the argument of $h_c(t)$ is a direct consequence of this notation, because the basepoints form a decreasing sequence.

Second and more important, the construction of a continuous impulse response enables a direct application of the hybrid analog/digital model introduced in Section 3.4.5 to analyze the properties of the resampling function.

If the discrete convolution process (3.96) is finite, then $h_c(t)$ is compactly supported. The impulse response $h_c(t)$ is constructed by combining the filter coefficients $h(k, \mu)$ according to

$$h_c(t) = \begin{cases} h\left(k, \frac{t}{T_i} + t_k\right), & (\mu_{min} - t_k) T_i \leq t < (\mu_{max} - t_k) T_i \\ 0, & \text{otherwise} \end{cases} \quad \text{for } k = 0, 1, \ldots, N.$$

(3.97)

It is observed that the argument of the filter coefficients $h(k, \mu)$ is normalized to the input sampling period T_i. The construction of $h_c(t)$ from the functions $h(k, \mu)$ is depicted in Figure 3.19.

Conversely, the coefficients of the discrete filter $h(k, \mu)$ are obtained from the continuous impulse response as

$$h(k, \mu) = h_c([\mu - t_k] T_i) \qquad \text{for } k = 0, 1, \ldots, N.$$

(3.98)

Because $h_c(t)$ is constructed from $N+1$ segments of length $(\mu_{max} - \mu_{min}) T_i = T_i$ each, its support is compact and of length $(N+1) T_i$. The start and end values of the first and last segment,

respectively, can be determined by utilizing the optimal range of the intersample interval (3.84)

$$\left(\mu_{min} - t_0\right) T_i = \left(\frac{t_0 + t_N - 1}{2} - t_0\right) T_i = \left(\frac{t_0 + t_0 - N - 1}{2} - t_0\right) T_i = -\frac{N+1}{2} T_i \tag{3.99a}$$

$$\left(\mu_{max} - t_N\right) T_i = \left(\frac{t_0 + t_N + 1}{2} - t_N\right) T_i = \left(\frac{t_N + t_N + N + 1}{2} - t_N\right) T_i = \frac{N+1}{2} T_i. \tag{3.99b}$$

In this way, the impulse response $h_c(t)$ is nonzero only in the interval $[-\frac{N+1}{2} T_i, \frac{N+1}{2} T_i]$. That is, its support is symmetric with respect to zero regardless of the basepoint set used

$$h_c(t) = 0 \qquad \text{for } |t| > \frac{N+1}{2} T_i. \tag{3.100}$$

Again, this identity follows from the restriction of the intersample range to the central interval of the basepoint set.

3.5.4 Impulse Response Symmetries

As shown above, the support of $h_c(t)$ is always symmetric, provided that the intersample position is chosen according to the optimal range (3.84).

However, there are several reasons to use continuous-time impulse responses that are symmetric with respect to zero, e.g. [VS96a, VS07]

$$h_c(t) = h_c(-t) \qquad \text{for } t \in \mathbb{R}. \tag{3.101}$$

First, a parametric description of $h_c(t)$ requires fewer coefficients, thus resulting in more efficient design and analysis methods. Second, the symmetry often permits more efficient implementations.

Finally, due to the properties of a Fourier transform of real-valued and even-symmetric functions, the continuous-time frequency response $H_c(j\Omega)$ is also real-valued and symmetric. This results in two primary advantages. First, design and analysis in the frequency domain can be performed in the domain of the real numbers and only on one half of the real axis compared to the complex-valued representation for arbitrary impulse responses. Second, as manifested in the real-valued frequency response, the reconstruction of the continuous-time signal $x_c(t)$ using a symmetric impulse response does not introduce phase distortions.[2]

Conversely, there are very few reasons to use non-symmetric impulse responses. In ordinary discrete-time filter design, the main intention of design specifications with nonlinear phase is to reduce the group delay or the phase delay of the filter. However, in case of resampling filters, the delay of the filter is determined by the optimal intersample position range (3.84) and is not induced by symmetry conditions imposed on $h_c(t)$. Consequently, the motivation for discrete-time filters with reduced delay does not apply to continuous-time resampling filters.

In summary, resampling functions based on symmetric impulse responses are advantageous with no apparent drawbacks. In addition, symmetry with respect to zero eases design and analysis due to its real-valued and symmetric frequency response. For this reason, such symmetric impulse responses are used predominantly in the following.

---

[2]Note that this characterization does not include the resampling step of the hybrid analog/digital model, which may introduce additional phase distortions. For the subtleties of describing the phase response of SRC systems, see [Väl95a, Section 3.3.9].

3.5.5 Causality of Resampling Filters

As shown above, the quality of a resampling filter depends on the location of the intersample position μ with respect to the basepoints of the samples involved in the discrete convolution. In particular, optimal performance is achieved only if μ resides in the central interval of the basepoint set. This requirement is incorporated in the determination of the intersample position μ and the sample index n according to (3.85) and (3.86).

However, in order to enable causal implementations, these interrelations impose a further constraint on the sampling instants $t = mT_o$ of the output signal.

It is assumed that the index k denotes the most recent sample of the input signal $x[k] = x(kT_i)$, corresponding to the sampling time kT_i. Therefore, in order to ensure causality in the filtering operation (3.83), the sample index n, corresponding to the most recent sample used in the convolution sum, must fulfill

$$n \leq k. \tag{3.102}$$

Inserting the formula for the sample index (3.85) yields the largest admissible sampling instant as a function of the most recent input sample k

$$t \leq \left(k - \frac{N-1}{2} \right) T_i. \tag{3.103}$$

In this way, the output sequence $y[m] = y(mT_o)$ of a causal resampling process is delayed by at least $(N-1)/2$ samples with respect to the input sequence $x[k] = x(kT_i)$. This delay is typically referred to as the *implementation delay* or the *implementation latency* of the system. It is worth emphasizing that the delay value of $(N-1)/2$ input sampling periods is independent of the basepoint set and of any symmetry conditions, but follows solely from the restriction of the intersample position range to the central interval of the basepoint set (3.84).

For FD filtering, the requirement (3.103) can be implemented in two possible ways. First, equivalent to resampling, the nominal delay value D can be biased by the constant implementation delay of $(N-1)/2$ samples. An alternative solution is to subtract the implementation delay from the nominal delay value D. This modification is already embodied in the determination of the integer part of the fractional delay value D_{int} (3.93). In terms of this variable, the requirement of causality translates to

$$D_{int} \geq 0. \tag{3.104}$$

Application of (3.93b) yields the minimum delay value that can be synthesized by a causal FD element of order N operating in the optimal intersample position range

$$D \geq \frac{N-1}{2}. \tag{3.105}$$

3.5.6 Summary

In this section, a unified description for the discrete-time filtering process underlying most VFD and ASRC algorithms has been introduced. For this reason, the definition of several notations such as basepoint sets, the calculation of the intersample position, and the discrete-time sample indices involved in the convolution sum formed a major part. As an immediate benefit, these representations enable FD filtering and ASRC to be described in a uniform way.

However, the results of this section go beyond the scope of purely notational issues. The notion of basepoint sets proves to be beneficial in providing a uniform notation for the different specialized forms of the convolution sum and provides insight into the distinctions and special requirements of these forms. Moreover, this formalism enables several properties of discrete resampling filters to be characterized in general form. Examples include the optimal intersample position range, the implementation delay, or the benefits of resampling filters with symmetric continuous impulse responses.

3.6 Algorithms and Implementation Structures

In the Sections 3.3 and 3.4, variable fractional delay filtering and arbitrary sample rate conversion have been characterized as two different approaches for delaying discrete-time sequences by arbitrary values. Despite the differences between these approaches, the algorithms and implementation structures used for both approaches are identical in most cases. In this section, the most widely-used implementation structures are characterized.

3.6.1 The Farrow Structure

The Farrow structure, proposed in [Far88], is a variable digital filter structure that is widely used both in VFD filtering and ASRC applications. It is based on a FIR filter with coefficients $h(n, \mu)$ that can be varied by means of a control variable. In case of the Farrow structure, the intersample position μ is used as control variable, and the filter coefficients form a polynomial of μ

$$h(n, \mu) = \sum_{m=0}^{M} c_{mn} \mu^m \quad \text{for } n = 0, 1, \dots, N. \tag{3.106}$$

Here, N denotes the order of the FIR filter, M is the order of the polynomial, and c_{mn} are the coefficients of the polynomials $h(n, \mu)$. The coefficients c_{mn} are compactly represented as a coefficient matrix C of dimension $(M + 1) \times (N + 1)$

$$C = \begin{bmatrix} c_{00} & c_{01} & \cdots & c_{0N} \\ c_{10} & c_{11} & \cdots & c_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M0} & c_{M1} & \cdots & c_{MN} \end{bmatrix}. \tag{3.107}$$

The transfer function $H(z, \mu)$ is given by

$$H(z, \mu) = \sum_{n=0}^{N} h(n, \mu) z^{-n} \tag{3.108a}$$

$$= \sum_{n=0}^{N} \left(\sum_{m=0}^{M} c_{mn} \mu^m \right) z^{-n}. \tag{3.108b}$$

Exchanging the order of summation yields

$$H(z, \mu) = \sum_{m=0}^{M} \left(\sum_{n=0}^{N} c_{mn} z^{-n} \right) \mu^m \tag{3.109a}$$

$$= \sum_{m=0}^{M} C_m(z) \mu^m \tag{3.109b}$$

$$\text{with } C_m(z) = \sum_{n=0}^{N} c_{mn} z^{-n} \quad \text{for } m = 0, 1, \ldots, N. \tag{3.109c}$$

Here, $C_m(z)$ denotes a set of $M + 1$ FIR filters that are commonly referred to as the subfilters of the Farrow structure. The discrete-time frequency response of the filter structure is obtained by evaluating $H(z, \mu)$ on the unit circle

$$H\left(e^{j\omega}, \mu\right) = \sum_{m=0}^{M} C_m\left(e^{j\omega}\right) \mu^m. \tag{3.110}$$

Continuous-Time Impulse Response As stated in (3.106), the filter coefficients $h(k, \mu)$ are represented by polynomials

$$h(k, \mu) = \sum_{m=0}^{M} c_{mn} \mu^m \quad \text{for } n = 0, 1, \ldots, N.$$

Consequently, the continuous impulse response $h_c(t)$ of the Farrow structure according to (3.97) is a piecewise polynomial function

$$h_c(t) = \begin{cases} \sum_{m=0}^{M} c_{mn} \left(\dfrac{t}{T_i} + t_n \right)^m, & (\mu_{min} - t_n)T_i \leq t < (\mu_{max} - t_n)T_i \\ 0, & \text{otherwise} \end{cases} \quad \text{for } n = 0, 1, \ldots, N. \tag{3.111}$$

Introducing basis functions for piecewise polynomials that are zero outside a given interval

$$f(m, \mu) = \begin{cases} \mu^m, & \mu_{min} \leq \mu < \mu_{max} \\ 0, & \text{otherwise} \end{cases} \tag{3.112}$$

enables a compact representation of $h_c(t)$

$$h_c(t) = \sum_{n=0}^{N} \sum_{m=0}^{M} c_{mn} f\left(m, \frac{t}{T_i} + t_n \right) \tag{3.113}$$

The basis functions $f(m, \mu)$ are defined in the style of [VS96b, VS07] but are not identical. In the mentioned reference, the time t is used as input argument and is transformed internally. In contrast, the basis function (3.112) uses the intersample position normalized to the input

Operation	Additions	Multiplications	Total
FIR filtering	$(M+1)N$	$(M+1)(N+1)$	$(M+1)(2N+1)$
Horner's scheme	M	M	$2M$
Total	$MN+M+N$	$MN+2M+N+1$	$2MN+3M+2N+1$

Table 3.1: Computational effort for the Farrow structure to compute one output sample.

sampling period as argument. In addition, the basepoint set notation enables arbitrary forms of the convolution sum (3.96) as motivated in Section 3.5. The class of resampling filters defined by a piecewise polynomial continuous impulse response is commonly denoted as polynomial-based interpolation filters [VS96a, Ves99, VS07]. It is important to distinguish this class from polynomial interpolation, such as Lagrange interpolation, which is based on fitting a polynomial to a set of discrete points. While polynomial interpolation can be generally stated as a polynomial-based interpolation filter, the converse does not hold. Consequently, polynomial-based interpolation filters and thus the Farrow structure can be used to implement a more general class of resampling filters, including polynomial interpolation as a subset.

Computational effort   Using the Farrow structure, the calculation of one output sample requires the evaluation of $M+1$ FIR filters of order N and the evaluation of a polynomial of order M, which is efficiently performed using Horner's scheme. The computational costs are summarized in Table 3.1. A block diagram for the implementation of the Farrow structure is depicted in Figure 3.20.
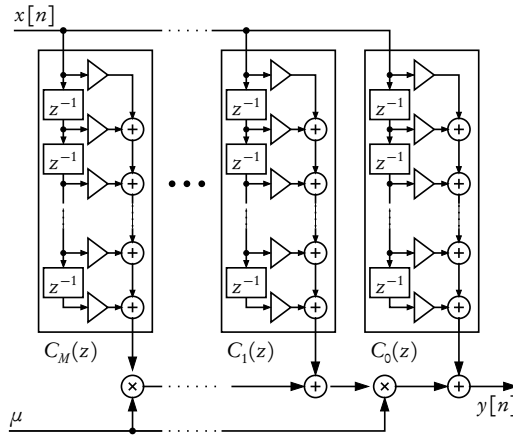
Figure 3.20: The Farrow structure.

3.6.2 The Modified Farrow Structure

A modification to the Farrow structure that reduces the computational complexity has been proposed in [VS96a]. This structure is commonly denoted the modified Farrow structure.

Although it has been initially proposed in the field of ASRC, this structure can be applied to FD filtering as well. The main idea underlying this structure is to enforce coefficient symmetries in the subfilters $C_m(z)$ to enable more efficient realizations of these filters.

3.6.2.1 Original Form of the Modified Farrow Structure

In the original representation of this structure [VS96a, Ves99, VS07], a convolution sum of the form (3.81) is used

$$
y(t) = \sum_{k=-\frac{N+1}{2}}^{\frac{N-1}{2}} x[n-k] h_c \left([\mu + k] T_i \right) \qquad \text{with} \quad 0 \le \mu < 1. \tag{3.114}
$$

Imposing symmetry conditions on the subfilters $C_m(z)$ makes the continuous impulse response $h_c(t)$ symmetric with respect to zero. For this purpose, two measures are taken. First, the control variable of the variable digital filter is made symmetric with respect to 0. Because the intersample position μ is used as control variable in the original Farrow structure, a transformation is introduced in [VS96a]

$$
\mu' = \frac{2t}{T_i} - 1 = 2\mu - 1. \tag{3.115}
$$

In this way, the range of the control variable is transformed from $0 \le \mu < 1$ to $-1 \le \mu' < 1$. This corresponds to the use of a modified basis function for the description of the piecewise polynomial impulse response $h_c(t)$ (3.113)

$$
f'(m, \mu) = \begin{cases} (2\mu - 1)^m, & 0 \le \mu < 1 \\ 0, & \text{otherwise} \end{cases}. \tag{3.116}
$$

The apostrophe is introduced here to distinguish between the basis functions for the original and the modified Farrow structure. In this way, the continuous impulse response for the modified Farrow structure results in

$$
h_c(t) = \sum_{n=-\frac{N+1}{2}}^{\frac{N-1}{2}} \sum_{m=0}^{M} c'_{mn} f'(m, \mu - n), \tag{3.117}
$$

where c'_{mn}, $0 \le m \le M, -\frac{N+1}{2} \le n \le \frac{N-1}{2}$ denote the elements of the coefficient matrix $C' = \{c'_{mn}\}$ of dimension $(M+1) \times (N+1)$.

The second measure to ensure the symmetry of the impulse response $h_c(t)$ is to impose symmetry and antisymmetry constraints on the elements c'_{mn}

$$
c'_{m(-n)} = \begin{cases} c'_{m(n-1)}, & m \text{ even} \\ -c'_{m(n-1)}, & m \text{ odd} \end{cases} = (-1)^m c'_{m(n-1)} \quad \text{for} \quad -\frac{N+1}{2} \le n \le \frac{N-1}{2}. \tag{3.118}
$$

In the original derivation [VS96a] and following works by these authors, the modified Farrow structure has been considered for odd filter orders N only. The even-order case has been investigated in [GBLR04, BGR04]. However, the discrete convolution sum used for the modified Farrow structure (3.114) is not applicable to even filter orders. Therefore, specific adaptations are required for the index range of the convolution as well as for the calculation of the sample index n and the intersample position μ.

3.6.2.2 Representation Using Symmetric Basepoint Sets

The representation of the modified Farrow structure described above requires an explicit transformation of the intersample range and thus complicates analysis and comparison to other resampling filters based on piecewise polynomial impulse responses. Furthermore, it is difficult to generalize this description to arbitrary, that is even and odd, orders of N. Using the notation of basepoint sets introduced in Section 3.5, it is possible to overcome these deficiencies.

The symmetric basepoint set defined in (3.89)

$$\{t_n\} = \frac{N}{2}, \frac{N}{2} - 1, \ldots, -\frac{N}{2}$$

corresponds to a symmetric range of the intersample position $-\frac{1}{2} \leq \mu < \frac{1}{2}$. Consequently, the control variable is a unit-sized interval as in case of the original Farrow structure. Therefore, it fulfills the required symmetry condition without an explicit transformation. Moreover, this symmetry holds for even and odd values of N. Using this basepoint set, the modified Farrow structure can be seen as a special case of the original Farrow structure that fulfills certain coefficient symmetries, but is compatible otherwise. For this reason, the apostrophe is omitted from the coefficient matrix $C = \{c_{mn}\}$ in order to distinguish it from $C' = \{c'_{mn}\}$ (3.117).

The elements c_{mn} are related to the coefficients c'_{mn} of the original form of the modified Farrow structure by

$$c_{mn} = 2^m c'_{m(n-\frac{N+1}{2})} \quad \text{for } 0 \leq m \leq M \text{ and } 0 \leq n \leq N. \tag{3.119}$$

This transformation accounts for the particular range of the index variable n in the representation (3.117) as well as for the translation of the control variable (3.115). It is noted that this transformation differs from the relation stated in [VS00, VS07] which describes the transformation between the original Farrow structure with an intersample range $0 \leq \mu < 1$ and the filter coefficients c'_{mn}.

In the symmetric basepoint form, the symmetry constraints of the modified Farrow structure (3.118) are represented by

$$c_{m(N-m)} = (-1)^m c_{mn} = \begin{cases} c_{mn}, & N \text{ even} \\ -c_{mn}, & N \text{ odd} \end{cases} \quad \text{for } 0 \leq m \leq M \text{ and } 0 \leq n \leq N \tag{3.120a}$$

Moreover, for even orders N, condition (3.120a) implies

$$c_{m\frac{N}{2}} = 0 \quad \text{for } N \text{ even and } m \text{ odd}. \tag{3.120b}$$

3.6.2.3 Discrete Frequency Response

As a result of the symmetry conditions (3.120), the subfilters $C_m(z)$ of the modified Farrow structure form linear-phase FIR filters. Specifically, for even m, $C_m(z)$ is of type I for even N and of type II for N odd. Likewise, for odd polynomial orders m, $C_m(z)$ has type III or IV for even or odd filter orders N, respectively. The condition (3.120b) coincides with the zero-valued central tap of a type III linear-phase filter.

For linear phase filters, alternative representations for the discrete frequency response

$$H(e^{j\omega}) = \sum_{n=0}^{N} c[n] e^{-j\omega}$$

exist [MP73, Sar93, OSB99]. A form suitable for the representation of the modified Farrow structure is

$$H(e^{j\omega}) = e^{-j\omega \frac{N}{2}} \sum_{n=0}^{N'} b[n] \, \mathrm{trig}(n,\omega) \,, \tag{3.121a}$$

where the trigonometric function $\mathrm{trig}(n,\omega)$ is defined by

$$\mathrm{trig}(n,\omega) = \begin{cases} 1, & \text{for type I, } n = 0 \\ 2\cos(n\omega), & \text{for type I, } n > 0 \\ 2\cos\left([n+\frac{1}{2}]\omega\right), & \text{for type II} \\ 2j\sin(n\omega), & \text{for type III} \\ 2j\sin\left([n+\frac{1}{2}]\omega\right), & \text{for type IV} \end{cases} \,. \tag{3.121b}$$

The coefficients $b[n]$ are related to the filter coefficients $c[n]$ by

$$b[n] = c[N'-n] \qquad \text{for } n = 0, 1, \ldots, N' \tag{3.122a}$$

$$c[n] = \begin{cases} b[N'-n], & 0 \le n \le N' \\ b[N'-N+n], & N'+1 \le n \le N \text{ for type I and II} \\ -b[N'-N+n], & N'+1 \le n \le N \text{ for type III and IV} \end{cases} \,, \tag{3.122b}$$

where N' determines the index range of $b[n]$

$$N' = \left\lfloor \frac{N}{2} \right\rfloor = \begin{cases} \frac{N}{2}, & \text{for type I and III } (N \text{ even}) \\ \frac{N-1}{2}, & \text{for type II and IV } (N \text{ odd}) \end{cases} \,. \tag{3.123}$$

This representation is similar to those found in standard signal processing texts, e.g. [Sar93, OSB99], but some modifications have been applied to make them more suitable for the modified Farrow structure. First, the range of the summation index $n = 0, 1, \ldots, N'$ is made identical for type I and III filters and for type II and IV filters, respectively. In this way, the expression $b[0]\sin(0\omega)$, which is invariably zero, is included in the expression for type III filters. However, this term has no practical impact because the filter coefficient $c[\frac{N}{2}]$ corresponding to $b[0]$ is restricted to zero by the antisymmetry conditions of type III linear-phase filters in any case.

Second, the constant phase shift of π for type III and IV filters is included as leading term j in the definition of the function $\text{trig}(n, \omega)$ (3.121b). In this way, the summation in (3.121a) does not denote the zero-phase part of the frequency response, as, for instance, in [PB87, Sar93]. Instead, the leading term $e^{-j\omega\frac{N}{2}}$ represents only the pure delay common to all types of causal linear-phase filters or, equivalently, the linear part of a system with generalized linear phase [OSB99].

Nonetheless, it is sensible to introduce the noncausal frequency response $H(\omega)$ of a linear-phase FIR filter

$$H\left(e^{j\omega}\right) = e^{-j\omega\frac{N}{2}} H(\omega) \qquad \text{with} \tag{3.124}$$

$$H(\omega) = \sum_{n=0}^{N'} b[n]\, \text{trig}(n, \omega)\,. \tag{3.125}$$

As argued above, $H(\omega)$ has to be distinguished from the zero-phase frequency response, although its notation is adopted from [Sar93].

Using this representation, the discrete frequency response of the modified Farrow structure is compactly expressed as

$$H(e^{j\omega}, \mu) = e^{-j\omega\frac{N}{2}} \sum_{m=0}^{M} C_m(\omega)\mu^m \qquad \text{with} \tag{3.126a}$$

$$C_m(\omega) = \sum_{n=0}^{N'} b_{mn}\, \text{trig}(m, n, \omega)\,, \tag{3.126b}$$

where $C_m(\omega)$ denotes the noncausal frequency response of the linear-phase subfilter $C_m(z)$. The function $\text{trig}(n, \omega)$ is extended by an additional argument for the polynomial order m

$$\text{trig}(m, n, \omega) = \begin{cases} 1, & \text{for } N \text{ even, } m \text{ even, } n = 0 \\ 2\cos(n\omega), & \text{for } N \text{ even, } m \text{ even, } n > 0 \\ 2\cos\left(\left[n + \frac{1}{2}\right]\omega\right), & \text{for } N \text{ odd, } m \text{ even} \\ 2j\sin(n\omega), & \text{for } N \text{ even, } m \text{ odd} \\ 2j\sin\left(\left[n + \frac{1}{2}\right]\omega\right), & \text{for } N \text{ odd, } m \text{ odd} \end{cases}. \tag{3.127}$$

Additionally, it is sensible to introduce the noncausal frequency response $H(\omega, \mu)$ of the modified Farrow structure that excludes the implementation delay of $N/2$ samples

$$H(\omega, \mu) = \sum_{m=0}^{M} C_m(\omega)\mu^m\,. \tag{3.128}$$

It is related to $H(e^{j\omega}, \mu)$ by

$$H(e^{j\omega}, \mu) = e^{-j\omega\frac{N}{2}} H(\omega, \mu)\,.$$

Due to the coefficient symmetries (3.120), the discrete frequency response of the modified Farrow structure is determined by the elements b_{mn} of a matrix B

$$B = \{b_{mn}\} \qquad \text{with } 0 \le m \le M \text{ and } 0 \le n \le N'\,, \tag{3.129}$$

(a) Type I (b) Type IV
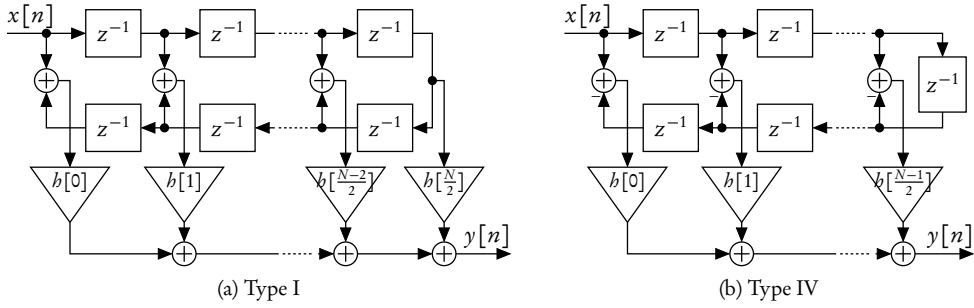
Figure 3.21: Efficient direct-form structures for linear-phase FIR filters.

where the order N' is defined by (3.123). The elements of B are related to the entries c_{mn} of the Farrow coefficient matrix C by

$$b_{mn} = c_{m(N'-n)} \qquad \text{for } 0 \leq m \leq M, \, 0 \leq n \leq N' \qquad (3.130a)$$

$$c_{mn} = \begin{cases} b_{m(N'-n)}, & 0 \leq n \leq N' \\ (-1)^m b_{m(N-N'+n)}, & N'+1 \leq n \leq N \end{cases} \qquad \text{for } 0 \leq m \leq M. \qquad (3.130b)$$

The antisymmetry condition for odd-order subfilters (3.120b) implies

$$b_{m0} = 0 \qquad \text{for } N \text{ even and } m \text{ odd}. \qquad (3.130c)$$

3.6.2.4 Continuous Impulse Response

The continuous impulse response of the modified Farrow structure is obtained by imposing the coefficient symmetries (3.120) on the impulse response $h_c(t)$ of the conventional Farrow structure (3.113). For the original form of the modified Farrow structure, an adapted form for $h_c(t)$ has been proposed in [VS96a, VS07]. By constructing modified basis functions from the basis functions $f'(m, \mu)$ (3.116)

$$g'(m,n,t) = \begin{cases} (-1)^m \left(2 \left[\frac{t}{T_i} + n + 1 \right] - 1 \right)^m, & -(n+1)T_i \leq t < -nT_i \\ \left(\left[2\frac{t}{T_i} - n \right] - 1 \right)^m, & nT_i \leq t < (n+1)T_i \\ 0, & \text{otherwise} \end{cases} \qquad (3.131a)$$

$$= (-1)^m f' \left(m, \frac{t}{T_i} + n + 1 \right) + f' \left(m, \frac{t}{T_i} - n \right), \qquad (3.131b)$$

the impulse response $h_c(t)$ can be expressed as

$$h_c(t) = \sum_{n=0}^{\frac{N-1}{2}} \sum_{m=0}^{M} c'_{mn} g'(m,n,t). \qquad (3.132)$$

In this representation, only the elements c'_{mn}, $0 \leq n \leq \frac{N-1}{2}$, $0 \leq m \leq M$ of the coefficient matrix C' are utilized.

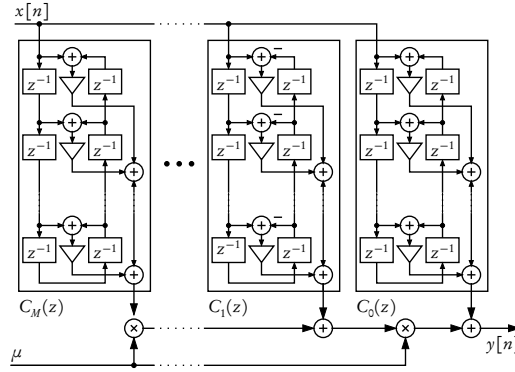| Filter type	Additions	Multiplications	Total
General FIR	N	$N+1$	$2N+1$
Type I	N	$\frac{1}{2}N+1$	$\frac{3}{2}N+1$
Type II	N	$\frac{1}{2}N+\frac{1}{2}$	$\frac{3}{2}N+\frac{1}{2}$
Type III	$N-1$	$\frac{1}{2}N$	$\frac{3}{2}N-1$
Type IV	N	$\frac{1}{2}N+\frac{1}{2}$	$\frac{3}{2}N+\frac{1}{2}$

Table 3.2: Computational effort for linear-phase FIR filters.

Figure 3.22: Modified Farrow structure for an odd filter order N. The subfilters $C_m(z)$ are linear-phase FIR filters of type II and IV for even and odd m, respectively.

3.6.2.5 Computational Effort

The symmetry relations fulfilled by the coefficients of linear-phase filters permit more efficient implementations compared to general FIR filters, essentially halving the number of required multiplications [OSB99]. Moreover, optimizations due to zero coefficient values are applicable in some cases. In Figure 3.21, implementation structures for type I and IV filter are shown exemplarily. Table 3.2 lists the operation counts compared to general FIR filters.

Applying these filter implementations enables efficient realizations of the modified Farrow structure. As an example, an implementation structure for N odd is shown in Figure 3.22. The operation counts for modified Farrow structures of arbitrary orders are summarized in Table 3.3. These performance figures do not assume coefficient symmetries or special values such as 0 or 1 in addition to the conditions (3.120).

3.6.3 Lagrange Interpolation

Lagrange interpolation is widely used in signal processing to determine signal values between discrete samples. It is applied both to FD filtering (see, among many others, [LVKL96, Väl95a, Smi10a, KPPV07]) and to sample rate conversion [SR73, EGH93, TKC94]. As a primary advantage, the coefficients of Lagrange interpolation filters can be calculated using explicit formulas. In this way, it is well suited for applications where the intersample position changes at

Operation	Order N	Additions	Multiplications
FIR filters	odd	$(M+1)N$	$(M+1)\frac{N+1}{2}$
	even	$(M+1)(N-1)+\lceil\frac{M+1}{2}\rceil$	$(M+1)\frac{N}{2}+\lceil\frac{M+1}{2}\rceil$
Horner's scheme		M	M
Total	odd	$(M+1)N+M$	$(M+1)\frac{N+1}{2}+M$
	even	$(M+1)(N-1)+\lceil\frac{M+1}{2}\rceil+M$	$(M+1)\frac{N}{2}+\lceil\frac{M+1}{2}\rceil+M$

Table 3.3: Computational effort for the modified Farrow structure to compute one output sample.

run time. As a second main advantage, the quality of Lagrange interpolators is very good at low frequencies, thus making it an ideal choice for audio applications such as physical modeling of musical instruments. If bandwidth of the signal is large, the quality of Lagrange interpolation deteriorates. However, for such applications, Lagrange interpolation may be used as a building block for more sophisticated algorithms. These structures are introduced in more detail in Section 3.6.5.

This section describes properties and implementation structures for Lagrange interpolation. Due to its widespread use, both directly and in combination with other techniques, the computational complexity of Lagrange interpolation is of major importance. Therefore, emphasis is put on the performance measures of a large number of implementation structures. The material presented in this section is mainly based on the article [Fra08].

3.6.3.1 Definition

Lagrange interpolation is one form of polynomial interpolation, a widely used method in numerical analysis. Given a set of function values x_n at $N+1$ distinct abscissas t_n, a unique polynomial of order N is constructed that passes through the points (t_n, x_n), $n = 0, 1, \ldots, N$. Evaluation of this polynomial enables interpolation at arbitrary values t in the range $[t_0, t_N]$. Due to the uniqueness of the interpolating polynomial, the results of different methods may differ only because of numerical inaccuracies.

Lagrange interpolation is a specific form of polynomial interpolation that is based on the evaluation of Lagrange polynomials [BT04]. However, in the field of signal processing, Lagrange interpolation and polynomial interpolation are often used synonymously. In the following, this definition will be adopted, and the term Lagrange interpolation will be used for all implementations that perform polynomial interpolation.

The Lagrange polynomials are defined as

$$l_k(t) = \frac{\prod_{i=0,i\neq k}^{N}(t - t_i)}{\prod_{i=0,i\neq k}^{N}(t_k - t_i)} \qquad \text{for } k = 0, 1, \ldots, N. \tag{3.133}$$

Interpolation based on these polynomials takes the form

$$y(t) = \sum_{k=0}^{N} l_k(t)x(t_k). \tag{3.134}$$

Consequently, Lagrange interpolation can be readily expressed as a discrete-time FIR filtering operation according to (3.83). In this way, the Lagrange polynomials $l_k(t)$ form the filter coefficients $h(k, \mu)$, while the basepoints t_k and the discrete samples $x[n-k]$ form the abscissas and ordinates of the data points determining the interpolation polynomial

$$y(t) = \sum_{n=0}^{N} x[n-k] h(k, \mu) \qquad \text{with} \tag{3.135a}$$

$$h(k, \mu) = \frac{\prod_{l=0, l \neq k}^{N} (\mu - t_l)}{\prod_{l=0, l \neq k}^{N} (t_k - t_l)} \qquad \text{for } k = 0, 1, \dots, N. \tag{3.135b}$$

3.6.3.2 Properties of Lagrange Interpolators

The Interpolation Condition   According to the definition of polynomial interpolation, Lagrange interpolation inherently fulfills the interpolation condition (3.33) at the data points used in the construction of the interpolating polynomial. In the notation of basepoint sets, this condition is expressed as

$$y(t_k) = x[n-k] \qquad \text{for } k = 0, 1, \dots, N. \tag{3.136}$$

This condition is equivalent to

$$h(n, t_k) = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases}. \tag{3.137}$$

The interpolation condition can be used to derive the Lagrange polynomials (3.135b), see, e.g. [Rut76, Väl95a, Fra08].

Maximal Flatness   The property of maximal flatness implies that the approximation error of a discrete filter of order N as well as its first N derivatives are zero at a predefined frequency ω_0 [Her92, Sar93, SAS04]. In order to obtain real-valued filter coefficients, ω_0 may be chosen either as $\omega_0 = 0$ or $\omega_0 = \pi$. For practical interpolators, $\omega_0 = 0$ is typically used [Her92, Väl95a], corresponding to good quality at low frequencies.

   The coefficients $h(n, \mu)$ can be derived using the condition of maximal flatness. In [Väl95a], this has been performed for FD filtering using the natural numbers as basepoint set (3.87). Here, we provide a derivation that holds for arbitrary basepoint sets.

   The condition for maximal flatness is formally stated as

$$0 = \frac{d^k}{d\omega^k} \left[H(e^{j\omega}, \mu) - \widehat{H}(e^{j\omega}, \mu) \right] \Big|_{\omega=0} \qquad \text{for } k = 0, 1, \dots, N \tag{3.138a}$$

$$0 = \frac{d^k}{d\omega^k} \left[\sum_{n=0}^{N} h(n, \mu) e^{j\omega t_n} - e^{j\omega \mu} \right] \Big|_{\omega=0} \qquad \text{for } k = 0, 1, \dots, N.. \tag{3.138b}$$

Note that no minus signs are present in the exponents in (3.138b) due to the use of the basepoints t_n and the intersample position μ, respectively. Differentiations and evaluation at $\omega_0 = 0$ results

in

$$\sum_{n=0}^{N} h(n,\mu)(jt_n)^k - (j\mu)^k = 0 \qquad \text{for } k = 0, 1, \ldots, N \tag{3.139a}$$

$$\sum_{n=0}^{N} h(n,\mu)t_n^k = \mu^k \qquad \text{for } k = 0, 1, \ldots, N. \tag{3.139b}$$

The equations for the $k = 0, 1, \ldots, N$ can be combined into a matrix equation

$$\begin{bmatrix} t_0^0 & t_1^0 & \cdots & t_N^0 \\ t_0^1 & t_1^1 & \cdots & t_N^1 \\ \vdots & \vdots & \ddots & \vdots \\ t_0^N & t_N^0 & \cdots & t_N^N \end{bmatrix} \cdot \begin{bmatrix} h(0,\mu) \\ h(1,\mu) \\ \vdots \\ h(N,\mu) \end{bmatrix} = \begin{bmatrix} \mu^0 \\ \mu^1 \\ \vdots \\ \mu^N \end{bmatrix}. \tag{3.140}$$

The matrix in (3.140) has the special structure of a Vandermonde matrix [Rut76, Rus89, PTVF92]. Since the basepoints t_k are distinct, it is guaranteed to be uniquely solvable. Moreover, due to its special structure, the solution of this matrix equation can be stated in explicit form

$$h(n,\mu) = \frac{\prod_{k=0,k\neq n}^{N}(\mu - t_k)}{\prod_{k=0,k\neq n}^{N}(t_n - t_k)} \qquad \text{for } n = 0, 1, \ldots, N.$$

In this way, the coefficients obtained from enforcing the property of maximal flatness are identical to the Lagrange polynomials (3.135b) resulting from the interpolation condition [MDRK93, Väl95a]. Consequently, resampling filters designed with respect to a maximal flatness criterion are identical to Lagrange interpolators.

Other notable properties of Lagrange interpolation are its equivalence to the sinc interpolator windowed by a binomial window [KW96, DM99, DM00]. The property of passivity [Väl95a, Smi10a], states that the magnitude response does not exceed unity when operated within the optimal intersample position interval. This characteristic is of especial value in applications where resampling algorithms are used in closed loops, because reliable bounds for the feedback gain can be utilized to ensure the stability of the system.

Numerical Properties   Because the results of polynomial interpolation are uniquely determined by the interpolating polynomial passing through the samples $x[n]$, different implementations may vary only due to numerical errors. In the field of signal processing, no results on the numerical accuracy of different implementation structures have been published so far.

In numerical analysis, numerical issues of polynomial interpolation in general and the Lagrange form in particular have been investigated extensively. The primary reasons of numerical errors are Runges phenomenon, that is, the tendency of the interpolating polynomial to oscillate between data points as the order N increases, and errors due to the accumulation of roundoff errors [BT04, Hig04]. In particular, equidistant basepoint locations, as generally used in signal processing applications, are characterized as prone to Runges phenomenon. Moreover, the Lagrange form of polynomial interpolation is shown to be sensitive to roundoff errors unless specialized forms of the Lagrange polynomials are used.

However, these results are not directly applicable to digital signal processing, which is concerned with bandlimited, equidistantly sampled signals and performs interpolation only in the central interval of the basepoint set. A numerical analysis that accounts for these properties is beyond the scope of this work.

3.6.3.3 Complexity Measures for Implementation Structures

To provide a meaningful comparison between different algorithms, a set of metrics has to be defined.

The number of additions and multiplications to compute one output sample is used as the primary measure for the computational complexity. For this purpose, calculation of the filter coefficients $h(n, \mu)$ and the actual filtering operation are combined to yield a single set of operation counts.

This corresponds to a mode of operation where the control variable μ potentially changes for every output sample. While for some applications, especially for FD filtering, a separation of these two steps might prove beneficial, it would hinder a uniform comparison because several implementation structures combine these steps into a single, inseparable operation. Moreover, an evaluation scenario where the intersample position changes with every output sample is well-suited for ASRC as well as for FD applications with continuously changing delays.

On modern superscalar or parallel architectures or for hardware platforms such as FPGA or VLSI architectures, the number of arithmetic operations is not sufficient to assess the suitability of an algorithm for implementation. For this reason, the *depth* of the algorithm, that is, the minimum number of sequential operations on an ideal parallel architecture [Ble96] is introduced as an additional performance measure. The depth does not only assess the performance on explicitly parallel platforms, but also for instruction level parallelism features of most contemporary processor architectures such as pipelining, superscalar execution or VLIW (very large instruction word) instruction sets.

The minimum number of sequential steps of an algorithm is also a measure for the degree of data dependency within an algorithm. If there are many dependencies, i.e., the depth is large, instruction-level parallelism cannot be utilized efficiently. This becomes evident, for instance, as pipeline stalls or, in case of superscalar or VLIW architectures, poor utilization of processor resources.

For the calculation of the depth, operations with more than two operands are modeled as balanced binary trees. In this way, an operation with N operands has a depth of $\lceil \log_2(N) \rceil$.

3.6.3.4 Implementations based on the Evaluation of Lagrange Polynomials

Direct evaluation of the Lagrange polynomials (3.135b) is widely used for performing Lagrange interpolation in signal processing [Väl95a, LVKL96, KPPV07].

In most implementations, a number of minor performance optimizations are applied [Väl95a, Str00]. The denominator of the Lagrange polynomial is precalculated as a scaling coefficient

$$c_k = 1/ \prod_{l=0, l \neq k}^{N} (t_k - t_l) \qquad \text{for } k = 0, 1, \ldots, N. \tag{3.141a}$$

The $N + 1$ distinct differences of the numerator terms are computed once to be reused in the calculation of different coefficients $h(k, \mu)$

$$d_k = \mu - t_k \qquad \text{for } k = 0, 1, \ldots, N. \tag{3.141b}$$

Using these quantities, calculation of the coefficients and filtering is performed as follows:

$$p_k = \prod_{l=0, l \neq k}^{N} d_l \qquad \text{for } k = 0, 1, \ldots, N \tag{3.141c}$$

$$h(k, \mu) = c_k p_k \qquad \text{for } k = 0, 1, \ldots, N \tag{3.141d}$$

$$y(t) = \sum_{k=0}^{N} h(k, \mu) x[n - k]. \tag{3.141e}$$

Including the filtering operation, computing one output sample requires $2N + 1$ additions and $N^2 + 2N + 1$ multiplications. The depth of the associated implementation structure is $2\lceil \log_2(N) \rceil + 2$. In [Väl95a, VH07], an optimization based on the reuse of two-term products $d_n d_{n+1}$, $n = 0, 1, \ldots, N - 1$ has been proposed. It reduces the number of operations to calculate the coefficients $h(n, \mu)$ to less than $N^2 + N$ operations.

3.6.3.5 Methods Based on Barycentric Lagrange Interpolation

In [Her92], an explicit formula for the coefficients of maximally flat FD filters has been derived. For Lagrange interpolation, which is a special case of a maximally flat FD filter with $\omega_0 = 0$, the filter coefficients $h(n)$ are determined as

$$h(n) = \beta \hat{h}(n) \qquad \text{for } n = 0, 1, \ldots, N \tag{3.142a}$$

$$\hat{h}(n) = \begin{cases} \delta(n - d), & d \text{ integer} \\ \frac{-1^n N!}{n!(N-n)!} \frac{1}{d-n}, & \text{otherwise} \end{cases} \tag{3.142b}$$

$$\beta = \begin{cases} 1, & d \text{ integer} \\ \frac{(-1)^N}{N!} \prod_{k=0}^{N} (d - k), & \text{otherwise} \end{cases}. \tag{3.142c}$$

Here, $\delta(n)$ denotes the discrete-time impulse function, and N is the order of interpolation.

The term β is efficiently computed as

$$\beta = \frac{1}{\sum_{n=0}^{N} \hat{h}(n)}. \tag{3.143}$$

In the general case, that is, if d is not an integral value, $3N + 1$ additions, $2N + 2$ multiplications, and $N + 2$ divisions are necessary for calculating the filter coefficients and evaluating one output sample. Additionally, testing whether d is an integer requires at least one comparison. The depth of the structure is $2\lceil \log_2(N + 1) \rceil + 4$.

Introducing weighting variables

$$w_n = (-1)^n \frac{N!}{n!(N - n)!}, \tag{3.144}$$

enables (3.142a) to be written as

$$h(n) = \frac{\frac{w_n}{n-d}}{\sum_{n=0}^{N} \frac{w_n}{n-d}} \, . \tag{3.145}$$

Thus, the FD filtering operation can be expressed as

$$y[n,d] = \frac{\sum_{k=0}^{N} \frac{w_k}{k-d} x[n-k]}{\sum_{k=0}^{N} \frac{w_k}{k-d}} \, . \tag{3.146}$$

In the field of numerical analysis, (3.146) is known as the *barycentric form* of Lagrange interpolation (sometimes denoted as the *second or true form of the barycentric formula*) [Rut76, BT04]. Barycentric Lagrange interpolation exhibits a number of favorable numerical properties. For instance, the complexity is linear with respect to the order for fixed basepoint sets, as in case of equidistantly sampled discrete-time signals. Additionally, the barycentric formula is numerically more stable than the classical Lagrange formulation, if optimal, non-equidistant basepoint sets are used [Hig04]. Nevertheless, as stated above, these numerical properties cannot be applied directly to the interpolation of equidistantly sampled, bandlimited signals.

In [HV08], a similar algorithm, termed *division-based method*, is described

$$h(n) = c_n \frac{\prod_{n=0}^{N} (d-n)}{d-n} \, , \tag{3.147}$$

where c_n denotes the scaling coefficient (3.141a). FD filtering based on formula (3.147) is equivalent to the *first form of the barycentric interpolation formula* [Rut76].

Using the notation of basepoint sets, resampling by means of barycentric Lagrange interpolation is expressed as

$$y(t) = \begin{cases} x[n-k], & \mu = t_k, \; k \in 0,1,\ldots,N \\ \frac{\sum_{k=0}^{N} \frac{\omega_k}{\mu - t_k} x[n-k]}{\sum_{k=0}^{N} \frac{\omega_k}{\mu - t_k}}, & \text{otherwise} \end{cases} \quad \text{with } \omega_k = \prod_{l=0,l\neq k}^{N} (t_k - t_l). \tag{3.148}$$

Both forms of barycentric Lagrange interpolation require a relatively low number of operations which is linear in N. Nevertheless, there is a number of drawbacks that limit their applicability, especially for signal processing applications.

First, on most processor architectures, especially on DSP platforms, divide operations are significantly slower than e.g., additions or multiplications. However, as argued in [HV08], efficient methods to approximate divisions can be applied.

Second, a comparison is required to test whether the intersample position coincides with a basepoint. In this case, the algorithm must take a different branch to avoid a division by zero. This conditional execution is likely to degrade the computational efficiency on many CPU architectures, especially on modern architectures with long execution pipelines. Moreover, it complicates parallel processing on SIMD *(Single Instruction, Multiple Data)* architectures.

It is notable that, for floating-point arithmetic, the barycentric interpolation formula remains stable if the intersample position is close to a basepoint t_k, resulting in a division by a very

small value. Although a denominator that converges to zero causes numerical problems in most cases, the special structure of the barycentric formula prevents divergence or loss of precision in the floating-point representation [Hen79, BT04, Hig04]. Therefore, an exact floating-point comparison can be used to determine whether d is integer.

3.6.3.6 The Farrow Structure for Lagrange Interpolation

Because the filter coefficients $h(n, \mu)$ are polynomials, Lagrange interpolation falls into the class of polynomial-based interpolation filters. Therefore, it can be implemented using the Farrow structure characterized in Section 3.6.1. This implementation structure has been proposed for Lagrange interpolation by many authors, including [EGH93, Väl95a, Den07a]. Because a Lagrange interpolator of polynomial order N consists of $N + 1$ coefficients $h(n, \mu)$, the order of the subfilters $C_m(z)$ coincides with the polynomial order. Thus, the coefficient matrix is a square $(N + 1) \times (N + 1)$ matrix.

The elements of the coefficient matrix C (3.107) can be determined from the Vandermonde matrix equation (3.140). In [Väl95a], this derivation is performed for FD filtering applications with a basepoint set corresponding to the natural numbers (3.87). Several methods have been proposed to adapt this coefficient matrix to other intersample position ranges such as $0 \leq \mu < 1$ or $-\frac{1}{2} \leq \mu < \frac{1}{2}$ [Väl95b, Väl95a, Den07b]. These methods use the coefficient matrix obtained for the natural numbers as basepoints and apply a matrix transformation to account for the shift of the intersample position range. In [Den09], numerical issues in applying this transformation are characterized and a numerically stable solution is proposed.

Here, a formula for the coefficient matrix is presented for arbitrary basepoint sets. This method is based on the interpolation condition (3.137)

$$h(n, t_k) = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \tag{3.149}$$

and the representation of the filter coefficients (3.106)

$$h(n, \mu) = \sum_{m=0}^{M} c_{mn} \mu^m \qquad \text{for } n = 0, 1, \ldots, N. \tag{3.150}$$

Stating the interpolation condition (3.149) for all $n = 0, 1, \ldots, N$ and all basepoints t_k, $k = 0, 1, \ldots, N$, results in a matrix equation

$$C \cdot V^T = I, \tag{3.151}$$

where C denotes the coefficient matrix C (3.107), I is a $(N + 1) \times (N + 1)$ identity matrix, and V^T is the transpose of the Vandermonde matrix introduced in (3.140)

$$V^T = \begin{bmatrix} t_0^0 & t_0^1 & \cdots & t_0^N \\ t_1^0 & t_1^1 & \cdots & t_1^N \\ \vdots & \vdots & \ddots & \vdots \\ t_N^0 & t_N^1 & \cdots & t_N^N \end{bmatrix}. \tag{3.152}$$
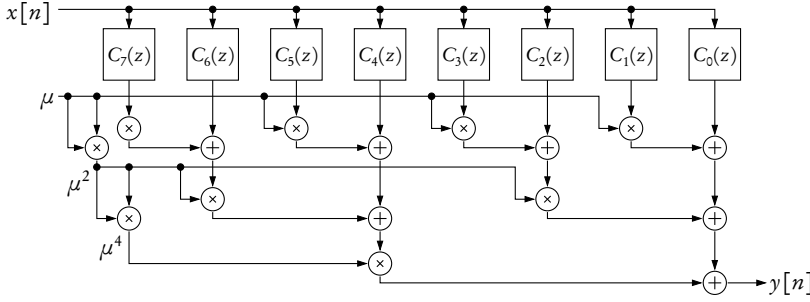
Figure 3.23: Farrow structure implementation for polynomial order $M = 7$ using Estrin's scheme for polynomial evaluation.

From the definition of the inverse matrix $A^{-1} \cdot A = I$, the coefficient matrix C follows directly

$$C = (V^T)^{-1}. \tag{3.153}$$

The existence and uniqueness of the solution is ensured because the basepoints t_k are distinct.

Computational Effort   For the Farrow structure applied to Lagrange interpolation, the computational effort follows from the performance figures of Table 3.1, with the polynomial order M being equal to the filter order N. In this way, $N^2 + 2N$ additions and $N^2 + 3N + 1$ multiplications are required to compute one output sample. The depth of the structure is $2N + \lceil \log_2(N+1) \rceil + 1$, that is, linear with respect to the interpolation order. This unfavorable property results from the sequential operation of Horner's scheme that combines the outputs of the subfilters.

However, there are alternatives to Horner's scheme that enable the parallel evaluation of polynomials at the expense of a slightly increased operation count, see, e.g. [AS89,Mul97,Knu98]. For instance, Estrin's scheme enables the evaluation of a polynomial of order N with a depth of $2\lceil \log_2(N+1) \rceil$ steps at the cost of $\lfloor \log_2(N) \rfloor$ additional multiplications to compute the powers μ^{2^k}, $k = 1, \ldots, \lfloor \log_2(N) \rfloor$.

In this way, the depth of the Farrow structure is reduced to $3\lceil \log_2(N+1) \rceil + 1$, and the number of multiplications is increased to $N^2 + 3N + \lfloor \log_2(N) \rfloor + 1$ per sample. In Figure 3.23, a Farrow structure of polynomial order $M = 7$ based on Estrin's scheme is shown.

3.6.3.7 The Modified Farrow Structure for Lagrange Interpolation

As stated in Section 3.6.2, symmetry relations in the coefficient matrix C can be utilized to implement the Farrow structure more efficiently. It has been shown that these symmetries correspond to a symmetric range of the control variable of the variable digital filter. In the original form of the modified Farrow structure [VS96a], this symmetric range is obtained by a transformation of the intersample position μ.

However, as derived in Section 3.5, a symmetric intersample position range $-\frac{1}{2} \leq \mu < \frac{1}{2}$ is also achieved if a symmetric basepoint set (3.89) is utilized. In this case, solving the matrix equation (3.153) yields a coefficient matrix C that fulfills the symmetry conditions (3.120). Consequently, Lagrange interpolation can be implemented by means of the modified Farrow structure if a symmetric basepoint set is used.

Computational Effort Basically, the number of additions and multiplications follows from the complexity of the modified Farrow structure displayed in Table 3.3 for $M = N$. However, for even orders N, the matrix exhibits a characteristic structure that can be utilized for a further reduction of the arithmetic complexity. As an example, the coefficient matrix for $N = 4$ is given by

$$C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} \\ -\frac{1}{24} & \frac{2}{3} & -\frac{5}{4} & \frac{2}{3} & -\frac{1}{24} \\ -\frac{1}{12} & \frac{1}{6} & 0 & -\frac{1}{6} & \frac{1}{12} \\ \frac{1}{24} & -\frac{1}{6} & \frac{1}{4} & -\frac{1}{6} & \frac{1}{24} \end{bmatrix}.$$

It is observed that the subfilter $C_o(z)$ consists of a single unity-valued central tap $c_{0(N/2)} = 1$ and is zero otherwise. This special structure results from the interpolation condition (3.137) applied to the parametric description of the filter coefficients of the Farrow structure (3.106). The zero-valued central taps for odd m follow from the symmetry conditions for the modified Farrow structure (3.120b). For these reasons, the number of additions and multiplications is reduced to $N^2 + N$ and $1/2 N^2 + 3/2 N$, respectively. For odd orders N, the coefficient matrix does not exhibit particular patterns compared to a general modified Farrow structure. Thus, the number of additions and multiplications for odd N amounts to $N^2 + 2N$ and $1/2 N^2 + 2N + 1/2$, respectively.

The depth of the implementation structure is identical to the original Farrow structure, that is, $2N + \lceil \log_2(N+1) \rceil + 1$. Likewise, the comments on reducing the depth by using parallel algorithms for the evaluation of a polynomial apply unaltered.

3.6.3.8 Farrow Implementation Structure for Lagrange Interpolation [Väl95b] for Lagrange interpolation

In [Väl95b], a transformation for the Farrow coefficient matrix for Lagrange interpolation has been proposed that enables a more efficient implementation. For N even, this transformation results in an intersample position range $-\frac{1}{2} \leq \mu < \frac{1}{2}$. Thus, the resulting structure is identical to the modified Farrow structure as described above. For odd N, the intersample position range is $0 \leq \mu < 1$, and the structure of C is more complex. As an example, the coefficient matrix for $N = 5$ reads

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{1}{30} & -\frac{1}{4} & 1 & -\frac{1}{3} & -\frac{1}{2} & \frac{1}{20} \\ 0 & -\frac{1}{24} & \frac{2}{3} & -\frac{5}{4} & \frac{2}{3} & -\frac{1}{24} \\ -\frac{1}{24} & \frac{7}{24} & -\frac{7}{12} & \frac{5}{12} & -\frac{1}{24} & -\frac{1}{24} \\ 0 & \frac{1}{24} & -\frac{1}{6} & \frac{1}{4} & -\frac{1}{6} & \frac{1}{24} \\ \frac{1}{120} & -\frac{1}{24} & \frac{1}{12} & -\frac{1}{12} & \frac{1}{24} & -\frac{1}{120} \end{bmatrix}.$$

First, the coefficients for $m = 0$ are zero except for $c_{m(N+1)/2} = 1$. Second, for higher even orders of m, the subfilter coefficient fulfill

$$c_{mn} = \begin{cases} 0, & n = 0 \\ c_{m(N - [n-1])}, & n = 1, 2, \ldots, N \end{cases}.$$
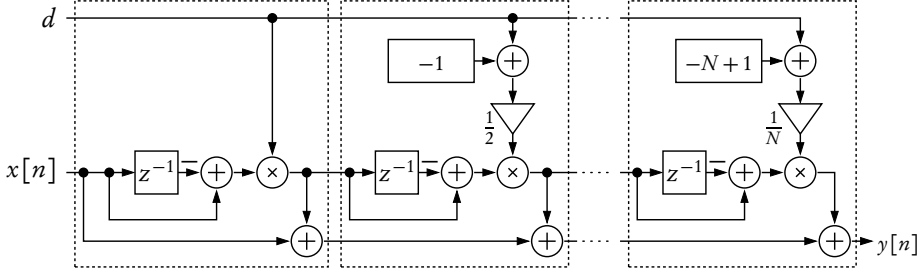
Figure 3.24: Modular linear-time implementation structure for Lagrange interpolation [TD97] for fractional delay filtering.

Consequently, the subfilters $C_m(z)$ can be implemented as type I linear-phase filter of order $N-1$ for m even, $m \geq 2$. For m odd, no specific symmetry relations hold.

Computational Effort   For N even, the structure and thus the computational cost is identical to the modified Farrow structure, thus the numbers of additions and multiplications are given by $N^2 + N$ and $1/2 N^2 + 3/2 N$, respectively. For N odd, the computational costs result from the structure of the coefficient matrix described above. $N^2 + 1/2 N + 1/2$ additions and $3/4 N^2 + 2N + 1/4$ multiplications are required to compute one output sample if N is odd.

The depth of the structure is identical to the original and the modified Farrow structure.

3.6.3.9 Implementation Structures based on Newton's Interpolation Formula

In [DT96, TD97], an implementation structure on the power series expansion of a Lagrange interpolation filter has been proposed. This structure is derived in the context of FD filtering. The interpolation filter $H_N(z, d)$ of order N

$$H_N(z,d) = \sum_{n=0}^{N} \frac{d(d-1)\cdots(d-[N-1])}{n!} \left(z^{-1} - 1\right)^{-n} \tag{3.154a}$$

can be stated in recursive form

$$H_N(z,d) = \begin{cases} 1, & N = 0 \\ H_{N-1}(z,d) + \dfrac{d(d-1)\cdots(d-[N-1])}{k!}(z^{-1}-1)^{-N}, & N > 0 \end{cases}. \tag{3.154b}$$

From this recursive representation, a modular implementation structure is derived by applying Horner's scheme to a polynomial consisting of powers of $(z^{-1} - 1)^{-1}$. The resulting structure is shown in Figure 3.24. An identical structure has been proposed in [Can07].

Equivalence to Newton's Interpolation Formula   In [LR09], the equivalence of this structure to Newton interpolation has been shown. Newton interpolation is an alternative form of
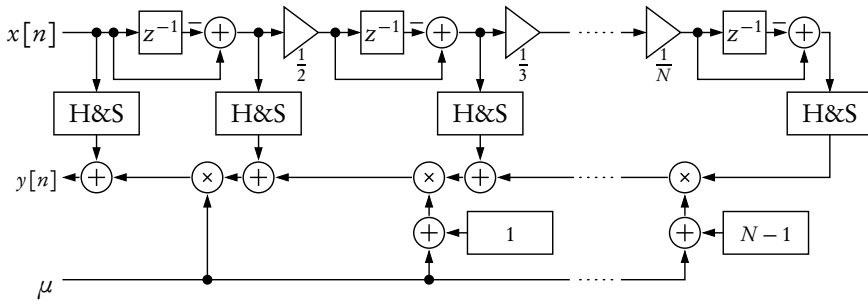
Figure 3.25: Modular linear-time structure for Lagrange interpolation for time-variant intersample positions [LR09].

polynomial interpolation, and thus functionally equivalent to Lagrange interpolation, [Rut76, WR66]. For equidistant sequences $x[n]$, it is given by

$$f(n + \mu) = \sum_{k=0}^{N} \frac{1}{k!} \nabla^k x[n] \prod_{l=0}^{k-1} (\mu + l), \qquad (3.155)$$

where ∇^k denotes the backward difference operator of order k

$$\nabla^k f(x) = \begin{cases} f(x), & k = 0 \\ \nabla^{k-1} f(x) - \nabla^{k-1} f(x-1), & k \geq 1 \end{cases}. \qquad (3.156)$$

It is easy to observe that the structure of Figure 3.24 implements Newton's interpolation formula. The components $1 - z^{-1}$ represent backward difference operators, and cascades of these blocks implement backward differences of higher orders.

Computational Effort   A single module consists of one delay element, three additions and two multiplications. In the first block, some operations are omitted as they represent trivial operations. In this way, computation of one output sample requires $3N - 1$ additions and $2N - 1$ multiplications. The depth of the structure is $2N - 1$, because the blocks of the modular implementation have to be processed sequentially.

Discussion   Despite this very low instruction count, this structure shows a number of deficiencies that severely limits its application for general-purpose FD and ASRC applications. First, the linear depth of $2N - 1$ hinders efficient parallel implementation or increases the latency of the system. Second, the state of the algorithm, represented by the values of the delay elements z^{-1}, depends on the current fractional delay value d or the intersample position μ in case of ASRC applications. In this way, the output samples depend on previous intersample positions, leading to transients if the fractional part d is changed.

Modification for Time-Variant Intersample Positions   Motivated by this deficiency, a modification of this structure, also based on Newton interpolation, has been proposed in [LR09].
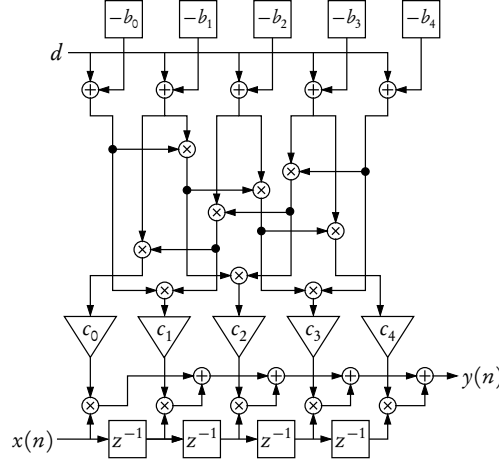
Figure 3.26: Linear-complexity implementation structure for Lagrange interpolation (Order $N = 4$).

Basically, the dependence on previous intersample positions is avoided by factoring the operations dependent on μ (or d) out of the delay-line-like structure containing the backward differences. The resulting structure is shown in Figure 3.25. A basically identical structure for FD filtering has been proposed independently in [CD10].

The H&S (hold and sample) elements decouple the processing for each input sample from the processing of an output value. Consequently, they are required only in case of ASRC algorithms where these operations occur asynchronously. For FD filters, that is, synchronous discrete-time systems, they can be replaced by simple lines.

The number of instructions is identical to the original form. However, the depth is increased to $4N - 1$, since the calculation of an output sample is also sequential and has to be performed after the computation of an input sample is completed.

For FD or ASRC applications that operate on variable-length delay lines with arbitrary access, both structures based on Newton's interpolation show an additional disadvantage. Irrespective of its dependency on the intersample position, the state contained in the delay elements z^{-1} of the backward differences depends on its position within this delay-line-like structure. Consequently, an arbitrary access which is required for instance for variable output times or multiple accesses to the same delay line is not possible.

3.6.3.10 Linear-Complexity Methods based on Lagrange Polynomials

An algorithm requiring a linear number of operations that is based on the evaluation of the Lagrange polynomials (3.135b) has been proposed in [MKT97]. Compared to the direct evaluation of these polynomials as described in Section 3.6.3.4, this complexity reduction is achieved by a systematic reuse of partial results that occur in the calculation of the product terms $p_k(\mu)$

(3.141c)

$$p_k(\mu) = \prod_{l=0, l \neq k} (\mu - t_l) \qquad k = 0, 1, \ldots, N \tag{3.157a}$$

$$= f_k \cdot r_k \qquad k = 0, 1, \ldots, N, \tag{3.157b}$$

where the variables f_k and r_k are referred to as *forward partial products* and *reverse partial products*, respectively. They are defined as

$$f_k = \prod_{l=0}^{k-1} (\mu - t_l) \quad = \prod_{l=0}^{k-1} d_l \tag{3.158a}$$

$$r_k = \prod_{l=k+1}^{N} (\mu - t_l) = \prod_{l=k+1}^{N} d_l, \tag{3.158b}$$

where d_l denotes the difference terms (3.141b). Based on these intermediate values, an algorithm for the computation of the filter coefficients $h(k, \mu)$ is constructed

$$f_0 = 1 \tag{3.159a}$$
$$f_k = f_{k-1} d_{k-1} \qquad \text{for } k = 1, \ldots, N \tag{3.159b}$$
$$r_N = 1 \tag{3.159c}$$
$$r_k = r_{n+1} d_{k+1} \qquad \text{for } k = N-1, \ldots, 0 \tag{3.159d}$$
$$p_k = f_k r_k \qquad \text{for } k = 0, \ldots, N \tag{3.159e}$$
$$h(k, \mu) = c_k \, p_k \qquad \text{for } k = 0, \ldots, N, \tag{3.159f}$$

where the scaling coefficients c_k are given by (3.141a). An implementation structure for this algorithm, including the discrete filtering, is shown in Figure 3.26. Basically the same algorithm has been proposed independently in [Str00].

Computational Effort Including the filtering operation, computing one output sample requires $2N + 1$ additions and $5N - 1$ multiply operations. Thus, the asymptotic complexity is $O(N)$, that is, linear with respect to the filter order. In contrast to the linear-complexity algorithms shown in Sections 3.6.3.5 and 3.6.3.9, this algorithm is suitable for general-purpose applications with time-varying intersample positions and does not require operations such as divisions or conditional execution. The depth of the implementation structure is $N + \lceil \log_2(N + 1) \rceil + 2$, that is, $O(N)$. This relatively large value is due to the sequential evaluation of the forward and reverse partial products (3.159b) and (3.159d) that is also observed in Figure 3.26.

3.6.3.11 A Parallel Linear-Complexity Implementation Structure

The linear-time implementation presented in the preceding section is highly sequential, thus limiting its efficiency on many hardware platforms. For this reason, we present a processing scheme for computing the product terms p_k (3.157a) that adopts the reuse of the partial products but permits a highly parallel computation. This structure has been initially proposed in [Fra08] and is derived in a formal way in [Fra10]. Here, the structure is introduced in an informal way only.

(a) General processing scheme

(b) Recursion for $P(M)$, M even

(c) Recursion for $P(M)$, M odd

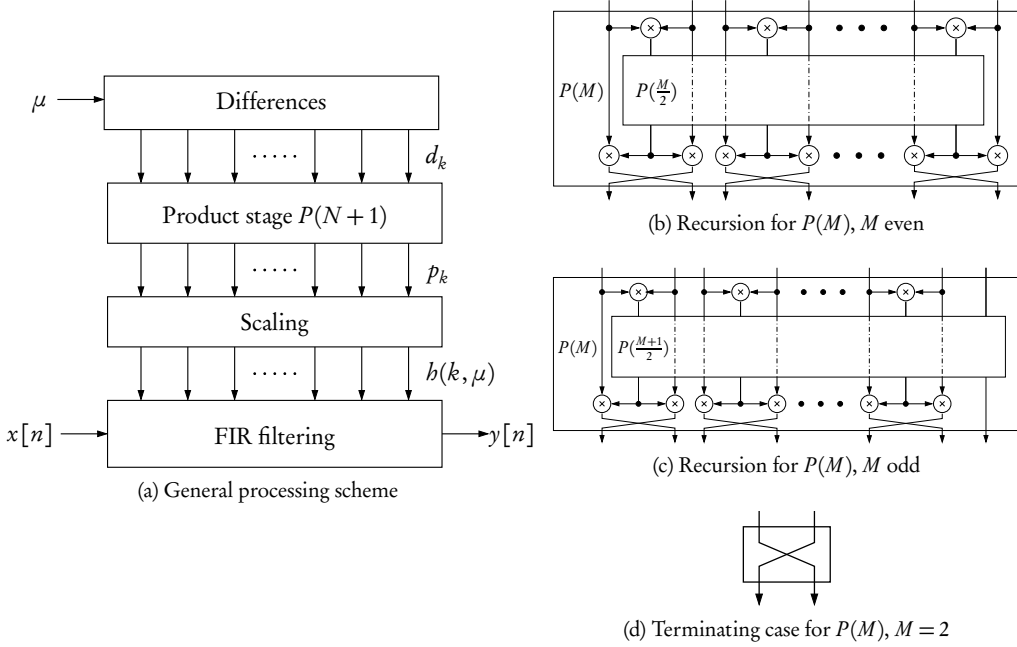(d) Terminating case for $P(M)$, $M = 2$

Figure 3.27: Parallel linear-complexity structure for Lagrange interpolation.

A similar algorithm which shares some characteristics with the linear-complexity structure characterized in Section 3.6.3.10 and some with the structure described here has been proposed independently in [HV08]. However, the emphasis on parallel evaluation is a unique feature of the parallel linear-complexity structure.

Derivation   Alike the linear-complexity structure described in Section 3.6.3.10, the proposed parallel structure can be partitioned into distinct stages as shown in Figure 3.27a. After calculating the difference terms d_k (3.141b), the $M = N + 1$ products p_k are computed in the product stage $P(M)$. Here, M denotes the number of in- and outputs. Scaling the results of this stage with the constants c_k (3.141a) yields the filter coefficients $h(k, \mu)$ (3.141d), which are used in a FIR filtering process to compute an output sample $y[n]$.

Thus, the distinction to existing methods lies in the product stage $P(M)$ that enables a highly parallel evaluation. For this purpose, the structure $P(M)$ is defined recursively as shown in Figure 3.27b. Pairs of adjacent inputs are multiplied and used as input for a product stage $P(\lceil M/2 \rceil)$ with $\lceil M/2 \rceil$ in- and outputs. The outputs of $P(\lceil M/2 \rceil)$ are multiplied with the respective inputs of $P(M)$ and exchanged pairwise.

If M is odd, then the last input is passed directly to the last input $P(\lceil M/2 \rceil)$. Likewise, the last output of $P(\lceil M/2 \rceil)$ is used as the respective output of $P(M)$ as shown in Figure 3.27c.

Using this recursion, the terminating case for $P(M)$ with $M = 2$ is reached after

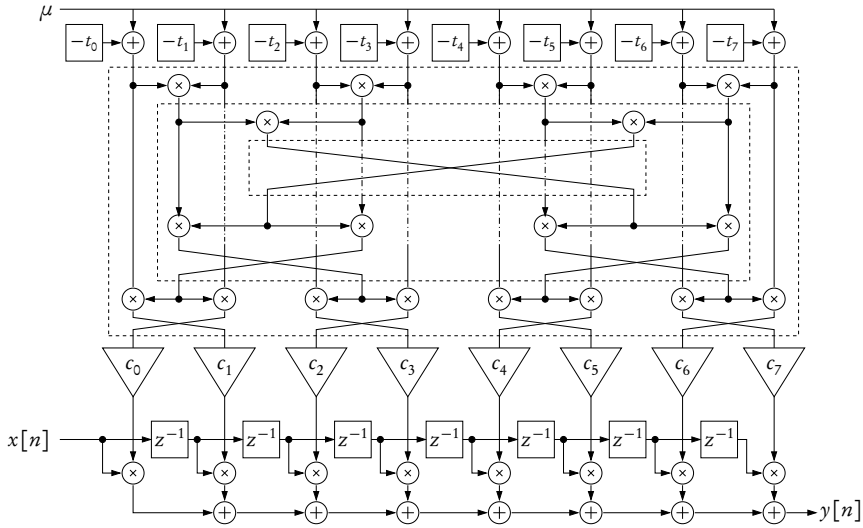$$K(M) = \begin{cases} 0, & M \leq 2 \\ \lceil \log_2(M) \rceil - 1, & M > 2 \end{cases} \qquad (3.160)$$

Figure 3.28: Parallel linear-complexity structure for Lagrange interpolation, order $N = 7$.

nontrivial stages. $P(2)$ performs an exchange of its two inputs as shown in Figure 3.27d and does therefore not require arithmetic operations.

The resulting structure for a complete Lagrange interpolation filter is shown in Figure 3.28 for $N = 7$.

Complexity analysis   The number of operations, that is, multiplications, in the product stage is denoted $M py(M)$ and follows from the recursive definition of $P(M)$

$$M py(M) = \begin{cases} 0, & M \leq 2 \\ \frac{3}{2}M + M py\left(\frac{M}{2}\right), & M \text{ even} , \\ \frac{3}{2}(M-1) + M py\left(\frac{M+1}{2}\right), & M \text{ odd} \end{cases} \tag{3.161}$$

which can be transformed by mathematical induction into an explicit expression

$$M py(M) = \begin{cases} 0, & M \leq 2 \\ 3(M-2), & M < 2 \end{cases}. \tag{3.162}$$

Combined with the other processing stages, the total number of instructions to compute one output sample amounts to $2N + 1$ additions and $5N - 1$ multiplications. Hence, it is identical to the sequential linear-complexity algorithm described in the preceding section. This confirms that the product stage is just an alternative way to compute the product terms p_k. That is, this stage performs basically the same operations as the sequential algorithm.

As apparent in Figures 3.27b and 3.27c, each nontrivial stage of the product stage requires two
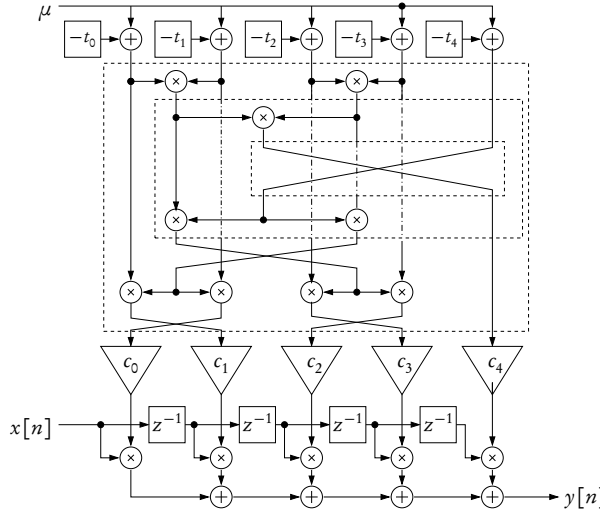
Figure 3.29: Parallel linear-complexity structure for Lagrange interpolation, order $N = 4$.

sequential operations. Thus, the depth $D(M)$ of the product stage $P(M)$ results in

$$D(M) = \begin{cases} 0, & M \leq 2 \\ 2K(M), & M > 2 \end{cases}. \tag{3.163}$$

where $K(M) = \lceil \log_2(M) \rceil - 1$ (3.160) denotes the number of stages. As a consequence, the depth of the complete structure including FIR filtering amounts to $3\lceil \log_2(N+1) \rceil + 1$.

A remarkable characteristic of this structure is that the complexity is a linear and thus strictly monotonic function of the interpolation order N. Although the multiplier structure is derived by a recursive bisection, its complexity does not degrade if the problem size, that is, the number of operands $M = N + 1$, is not a power of two or a highly composite number. This property is in contrast to most algorithms based on a divide-and-conquer strategy, for instance many FFT algorithms.

Contrary to divide-and-conquer algorithms, which divide a problem into two or more smaller subproblems, the proposed multiplier structure transforms a problem into a single subproblem. This strategy is often referred to as *decrease-and-conquer*, e.g. [Lev07]. The linear complexity as well as its straightforward adaptation to arbitrary problem sizes are direct consequences of this strategy. To illustrate this adaptation, the implementation structure for a Lagrange interpolation filter of order $N = 4$, utilizing a multiplier stage with $M = 5$ inputs, is shown in Figure 3.29.

Complexity Reduction Using Coefficient Symmetries   The performance of the proposed structure can be further improved by exploiting symmetries in the scaling coefficients c_k (3.141a)

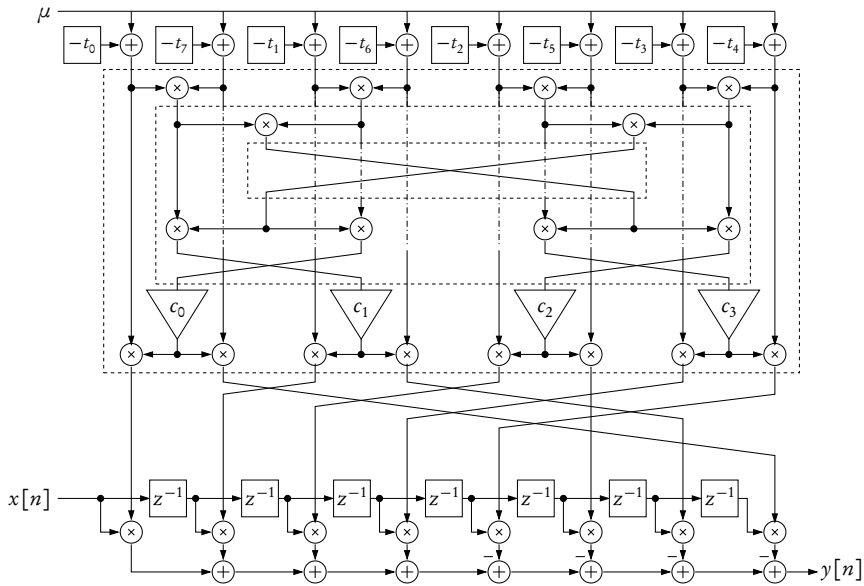$$c_k = 1 / \prod_{\substack{l=0 \\ l \neq k}}^{N} (t_k - t_l) \qquad \text{for } k = 0, 1, \dots, N.$$

Figure 3.30: Parallel linear-complexity structure for Lagrange interpolation with reduced scaling effort, order $N = 7$.

In [Fra08, Appendix 2], a relation between the coefficients c_k and c_{N-k} is derived which holds for arbitrary basepoint sets

$$c_{N-k} = \begin{cases} c_k, & N \text{ even} \\ -c_k, & N \text{ odd} \end{cases} \qquad \text{for } k = 0, 1, \ldots, N. \tag{3.164}$$

If the difference terms d_k, which form the inputs to the multiplier structure, are grouped according to this coefficient symmetry, the scaling operation can be shifted before the final multiplication in the stage $P(N+1)$. The resulting structure is shown in Figure 3.30. In this way, the scaling effort is reduced to $\lceil \frac{N+1}{2} \rceil$ operations, resulting in an overall number of $4N + \lceil \frac{N+1}{2} \rceil - 1$ multiplications. For odd filter orders N, half of the additions in the FIR filter have to be replaced by subtractions due to the antisymmetry of (3.164). However, it is assumed that this change can be incorporated into the FIR filter without additional effort in most applications. The number of additions as well as the depth of the structure is not affected by this optimization.

3.6.3.12 Performance Comparison

The instruction counts and the depth for the implementation structures considered in this section are shown in Figure 3.31. The corresponding formulas are summarized in Table 3.4.

With respect to the number of instructions, the algorithms fall into two categories. While the method based on the evaluation of the Lagrange polynomials and all variants of the Farrow structure have a complexity of $O(N^2)$, the other methods show a growth linear to the filter order.

(a) Number of additions

(b) Number of multiplications

(c) Number of instructions

(d) Depth

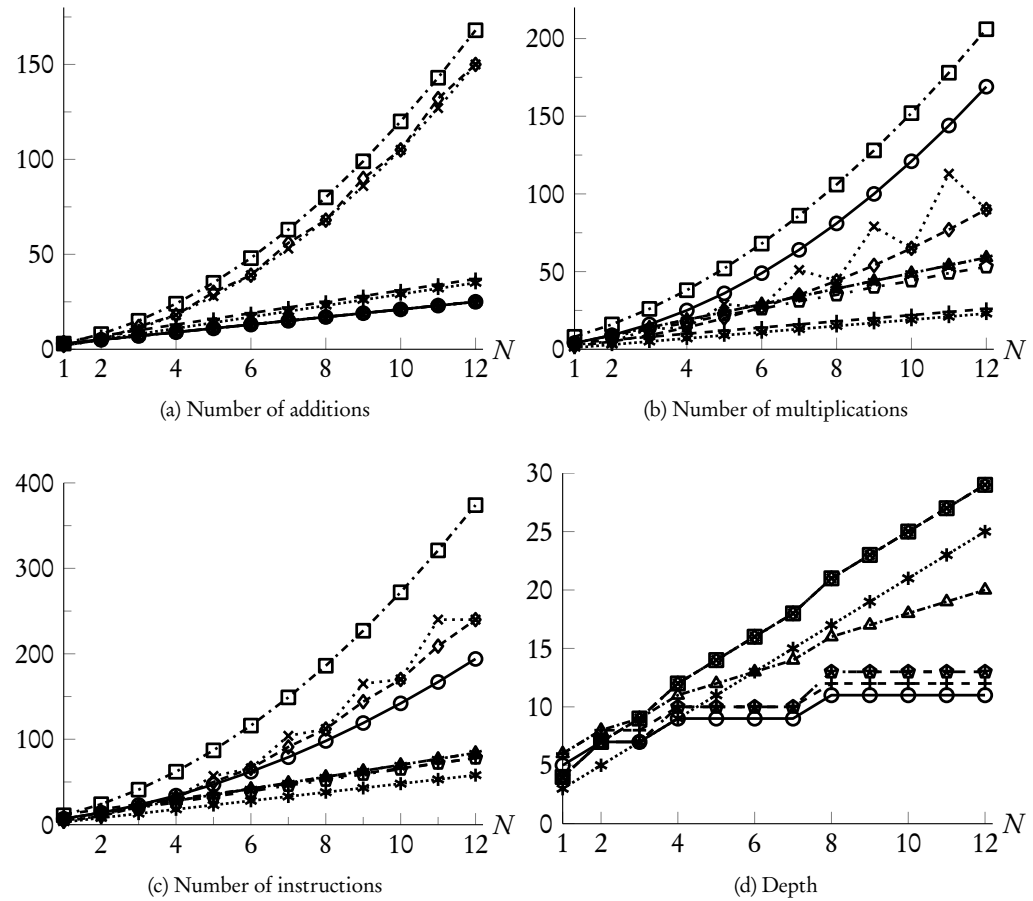| Algorithm	Section	Label
Classical formula	3.6.3.4	—⊖—
Barycentric formula	3.6.3.5	– – + – –
Farrow Structure	3.6.3.6	– ·⊡· –
Modified Farrow structure	3.6.3.7	····×····
Farrow implementation structure	3.6.3.8	– –◇– –
Newton's interpolation formula	3.6.3.9	······✱······
Sequential linear-complexity structure	3.6.3.10	– ··△··–
Parallel linear-complexity structure	3.6.3.11	– ✳ – ·
Parallel linear-complexity structure, reduced scaling effort	3.6.3.11	– ·⬠· –

Figure 3.31: Computational effort of implementation structures for Lagrange interpolation.

Algorithm	Section	N	Additions	Multiplications	Depth	Remarks
Classical Formula	3.6.3.4		$2N+1$	N^2+2N+1	$2\lceil\log_2(N+1)\rceil+2$	
Barycentric Lagrange Interpolation	3.6.3.5		$3N+1$	$2N+2$	$2\lceil\log_2(N+1)\rceil+4$	[a]
Farrow structure	3.6.3.6		N^2+2N	N^2+3N+1	$2N+\lceil\log_2(N+1)\rceil+1$	
Modified Farrow structure	3.6.3.7	odd	N^2+2N	$\tfrac{1}{2}N^2+2N+\tfrac{1}{2}$	$2N+\lceil\log_2(N+1)\rceil+1$	
		even	N^2+N	$\tfrac{1}{2}N^2+\tfrac{3}{2}N$	$2N+\lceil\log_2(N+1)\rceil+1$	
Farrow implementation structure [Väl95b]	3.6.3.8	odd	$N^2+\tfrac{1}{2}N+\tfrac{1}{2}$	$\tfrac{3}{4}N^2+2N+\tfrac{1}{4}$	$2N+\lceil\log_2(N+1)\rceil+1$	
		even	N^2+N	$\tfrac{1}{2}N^2+\tfrac{3}{2}N$	$2N+\lceil\log_2(N+1)\rceil+1$	
Newton's interpolation formula	3.6.3.9		$3N-1$	$2N-1$	$2N+1$	
Sequential linear-complexity structure	3.6.3.10		$2N+1$	$5N-1$	$N+\lceil\log_2(N+1)\rceil+2$	
Parallel linear-complexity structure	3.6.3.11		$2N+1$	$5N-1$	$3\lceil\log_2(N+1)\rceil+1$	
Parallel linear-complexity structure, symmetric	3.6.3.11		$2N+1$	$4N+\lceil N/2\rceil-1$	$3\lceil\log_2(N+1)\rceil+1$	

Table 3.4: Complexity of implementation structures for Lagrange interpolation.

[a] Additionally, $N+1$ divisions and one comparison are required

Among these, the methods based on barycentric Lagrange interpolation require special operations such as divisions or conditional execution, which are costly on many target architectures. The methods based on Newton's interpolation formula exhibit an extremely low instruction count, but are not suited for general-purpose applications such as variable-length delay lines with arbitrary access. In addition, the large depth value, that is, the number of sequential steps required on a parallel architecture, makes these algorithms ill-suited for parallel, superscalar or pipelined architectures. While variants of this algorithm that enable transient-free variable delays have been proposed recently, these modifications increase the depth of the structures further. The sequential linear-time algorithm described in Section 3.6.3.10 shows a relatively low complexity and is suited for general-purpose arbitrary delay operations. However, the depth linear to the filter deteriorates the efficiency on architectures with explicit or implicit parallelism. The proposed parallel linear-complexity implementation structure is based on this sequential algorithm, but overcomes the strong sequential dependencies. Moreover, this algorithm can be optimized further by exploiting coefficient symmetries, which is not possible for the sequential linear-complexity algorithm. In this way, this proposed structure forms a sensible choice for general-purpose applications on many contemporary platforms, including explicitly parallel, but also superscalar, pipelined, or SIMD architectures.

A key message of this section is that there exist several linear-complexity algorithms for Lagrange interpolation suitable for general-purpose FD or ASRC applications. Therefore, algorithms with complexity $O(N^2)$, such as all variants of the Farrow structure, should not be considered as an efficient way to implement Lagrange interpolation even for low or moderate filter orders. Furthermore, Lagrange interpolators can be implemented more efficiently than general resampling filters of comparable order. In this way, Lagrange interpolation is an interesting candidate for many VFD and ASRC applications, in particular if it is utilized as a building block within more sophisticated algorithms.

3.6.4 Splines for Signal Processing

Spline functions are a fundamental tool in numerical analysis and many other applications of digital computers, for instance computer graphics or computer aided design. The basic purpose of splines is to interpolate or approximate a sequence of function values by a smooth function that is composed of piecewise polynomials. Thus, it is well suited for the delay interpolation problems considered here. The degree of smoothness, which is quantified by the number of continuous derivatives of the interpolating or approximating function, can be controlled by the order of the spline function and other parameters of the particular method in use.

The work of Hou and Andrews [HA78] marks the first influential use of splines in signal processing. Further developments [UAE91, AUE92, UAE93a, UAE93b] proposed spline interpolation in the framework of digital filtering. [Uns99] provides an extensive survey of splines in the context of digital signal processing. However, this research is mainly focused on image processing, and the use of splines for other signal processing applications has been relatively limited.

In audio signal processing, use of splines for fractional delay filtering or resampling has been evaluated for instance in [ZB94] and [WBJ99]. In [LVKL96], splines are considered non-optimal from the frequency-domain viewpoint. For arbitrary sample rate conversion, spline interpolators are, among others, investigated in [Eva00b] and [VS07]. Again, they are primarily regarded

as time-domain interpolation methods that lack frequency-domain properties as required for general-purpose signal processing applications.

3.6.4.1 Spline Representations of Functions

The concept of splines has been introduced by I.J. Schoenberg [Sch46a, Sch46b] to interpolate or approximate smooth functions from discrete data. In its most general form, a spline function is represented by [Sch46a, Uns99]

$$y(t) = \sum_{-\infty}^{\infty} c[n] \beta^N(t-n), \tag{3.165}$$

where $c[n]$ denotes a set of coefficients and $\beta^N(t)$ is a basis function of order N.

In this original form, splines are derived for equidistant data points. While subsequent developments in the field of numerical analysis often focus on arbitrary, possibly coinciding sampling instants (see, e.g. [Sch73, dB76, Boo01b, Sch07]), the equidistant case (3.165) is most appropriate for use in discrete-time signal processing, where discrete sequences are typically associated with equidistantly sampled functions.

In (3.165), the sequence $c[n]$ denotes the spline coefficients (or B-spline coefficients [UAE93a, Uns99]), which are computed from, but are distinct from, the discrete-time signal $x[n]$ to be approximated. In this way, spline interpolation or approximation is invariably a two-stage process: In the first stage, a discrete set of coefficients is calculated, while in the second stage, these coefficients are used to reconstruct the function at arbitrary instants. However, if the first stage is left out, and the sample values $x[n]$ are used as spline coefficients instead, the quality of the resampling process deteriorates. In fact, this omission led to numerous misleading conclusions about the quality of spline methods for signal processing. In [AUE92, UAE91], errors due to this misconception are reported in the field of image processing, where omitting the coefficient calculation step leads to increased image blurring. For audio signal processing, [ZB94, WBJ99, PVN+10] report a roll-off of the frequency response towards higher frequencies, which is in fact due to the missing transformation into spline coefficients.

3.6.4.2 Spline Basis Functions

The terms $\beta^N(t)$ in equation (3.165) are referred to as the basic functions, basic spline functions [Sch46a], fundamental spline functions [CS66] or B-splines [dB76, Uns99, Mei02]. Because the use of terms "B-spline" or "B-spline interpolation" apparently led to considerable confusion, $\beta^N(t)$ will be referred to as "spline basis function" or "B-spline basis function" exclusively in the following. Spline basis functions $\beta^N(t)$ of order N are piecewise polynomial functions of polynomial order N which are continuous and $N-1$ times continuously differentiable. Being linear combinations of spline basis functions, spline functions according to (3.165) are consequently also smooth and $N-1$ times continuously differentiable.

In [Sch46a], the spline basis functions are introduced as

$$\beta^N(t) = \frac{1}{N!} \delta^{N+1} t_+^N, \tag{3.166}$$

where δ^k denotes the k-th order central difference operator with unit step size [WR66, AS65]

$$\delta^k f(x) = \begin{cases} f(x), & k = 0 \\ \delta^{k-1} f\left(x + \frac{1}{2}\right) - \delta^{k-1} f\left(x - \frac{1}{2}\right), & k > 0 \end{cases}. \tag{3.167}$$

It is notable that in this definition, the locations of the function values are not adjusted to integer displacements $f(x + n)$, $n \in \mathbb{Z}$ of the argument as, for instance, in [AS65]. Therefore, $f(x)$ is evaluated at integral values of x for k even, but at positions halfway between the integers, that is, $x = n + \frac{1}{2}$, $n \in \mathbb{Z}$, for k odd. The recursive definition (3.167) can be stated in explicit form as

$$\delta^k f(x) = \sum_{n=0}^{k} (-1)^n \binom{k}{n} f\left(x + \frac{k}{2} - n\right). \tag{3.168}$$

The term t_+^N of equation (3.166) denotes the one-sided power function [Uns99] or truncated power basis [Boo01b]

$$x_+^n = \begin{cases} x^n, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{3.169a}$$

$$= H(x) x^n, \tag{3.169b}$$

where the latter form utilizes the Heaviside step function or unit step function $H(x)$ [Bra00]

$$H(x) = \begin{cases} 0, & x < 0 \\ \frac{1}{2}, & x = 0 \\ 1, & x > 0 \end{cases}. \tag{3.170}$$

Applying the explicit form of the central difference operator (3.168) to (3.166) yields an explicit expression for the B-spline basis function of order N [Uns99]

$$\beta^N(t) = \frac{1}{N!} \sum_{n=0}^{N+1} (-1)^N \binom{N+1}{n} \left(t + \frac{N}{2} - n\right)_+^N. \tag{3.171}$$

The spline basis functions for $N = 0, \ldots, 3$ are shown in Figure 3.32. It is noted that $\beta^N(t)$ represent the central or symmetric spline basis functions because they are symmetric with respect to $t = 0$. While other definitions, such as causal splines that are non-zero only for $t \geq 0$ (e.g. [Sch73, BTU01]), are used in literature, symmetric B-spline basis functions are used exclusively in this work. In this way, the advantageous properties of resampling functions with symmetric impulse responses described in Section 3.5.4 apply.

Symmetric spline basis functions exhibit several noteworthy properties. They are real functions and are nonnegative on the complete real axis. Moreover, they have compact support. In particular, they are non-zero only in the interval $[-\frac{N+1}{2}, \frac{N+1}{2}]$. Moreover, as the functions $\beta^N(t)$ are $N - 1$ times continuously differentiable, they belong to the class

$$\beta^N(t) \in C^{N-1}, \tag{3.172}$$

(a) $\beta^0(t)$ (b) $\beta^1(t)$ (c) $\beta^2(t)$ (d) $\beta^3(t)$
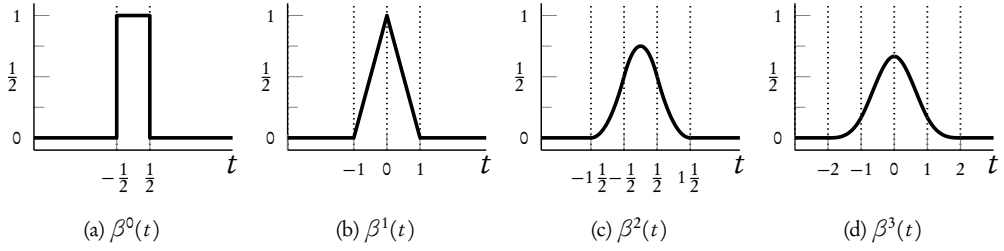
Figure 3.32: Symmetric spline basis functions for orders $N = 0, \ldots, 3$.

where C^k denotes the class of functions that are k times differentiable with a continuous k-th derivative (e.g. [Pow91]). This follows from $x_+^N \in C^{N-1}$ and the linearity of the central difference operator [Nør24]. Moreover, the N-th derivative of $\beta^N(t)$ is discontinuous only at the junctions between the polynomial segments, which are commonly referred to as the knots of the spline (e.g. [Sch73, UAE91, Uns99]). For symmetric B-splines, the knots are located at

$$t = n - \frac{N+1}{2} \qquad \text{for } n = 0, 1, \ldots, N+1, \tag{3.173}$$

that is, they lie on integer values of t for odd N while they reside halfway between the integers for even N.

As observed in Figure 3.32a, the spline basis function of order 0 is a unit rectangle function, which is defined as [Bra00]

$$\Pi(x) = \begin{cases} 1, & |x| < \frac{1}{2} \\ \frac{1}{2}, & |x| = \frac{1}{2} \\ 0, & |x| > \frac{1}{2} \end{cases} \tag{3.174a}$$

or, equivalently, using the unit step function (3.170)

$$= H\left(x + \tfrac{1}{2}\right) - H\left(x - \tfrac{1}{2}\right). \tag{3.174b}$$

Here, $H(x)$ denotes the Heaviside step function (3.170). The continuous Fourier transform of $\Pi(t)$ is given by

$$\mathscr{F}\{\Pi(t)\} = \operatorname{sinc}\left(\frac{\omega}{2}\right). \tag{3.175}$$

Thus, the frequency response of $\beta^0(t)$ is explicitly expressed as

$$B^0(\omega) = \mathscr{F}\left\{\beta^0(t)\right\} = \operatorname{sinc}\left(\frac{\omega}{2}\right), \tag{3.176}$$

where $\operatorname{sinc}(x)$ denotes the unnormalized Sinc function (3.29). For higher orders, $\beta^N(t)$ is formed by repeated convolution of $\beta^0(t)$ [HA78, Hec86, Uns99]

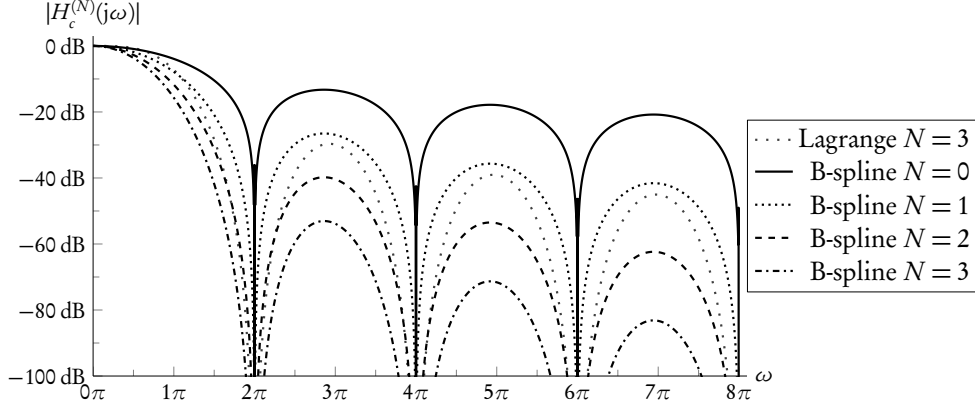$$\beta^N(t) = \underbrace{\Pi(t) * \Pi(t) * \cdots * \Pi(t)}_{N+1} \tag{3.177a}$$

Figure 3.33: Continuous frequency responses of spline basis functions, orders $N = 0, \ldots, 3$.

$$= \beta^{N-1}(t) * \Pi(t) \tag{3.177b}$$

$$= \int\limits_{-\infty}^{\infty} \beta^{N-1}(t - \tau)\Pi(\tau)\,\mathrm{d}\tau \tag{3.177c}$$

$$= \int\limits_{-\frac{1}{2}}^{\frac{1}{2}} \beta^{N-1}(t - \tau)\,\mathrm{d}\tau. \tag{3.177d}$$

Because convolution in the time domain corresponds to multiplication in the frequency domain, the Fourier transform of $\beta^N(t)$ for arbitrary orders N follows from (3.177)

$$B^N(\omega) = \mathscr{F}\left\{\beta^N(t)\right\} = \operatorname{sinc}\left(\frac{\omega}{2}\right)^{N+1}. \tag{3.178}$$

 In Figure 3.33, the frequency responses of spline basis functions of orders $N = 0, \ldots, 3$ are shown. It is observed that the asymptotic image attenuation increases monotonically with the order N. For example, for $N = 3$, it is significantly higher than for a Lagrange interpolator of the same order. On the other hand, the passband roll-off of spline basis functions is larger than that of Lagrange interpolation and increases for larger orders.

    It is remarkable that the transform-domain properties of spline basis functions played a central role in the development of spline interpolation and approximation [Sch46a, Sch46b]. Although the generic mathematical framework developed there does not have a direct interpretation in terms of a frequency response, the Fourier transform of a continuous function describing an interpolation or approximation method is termed the characteristic function of this method. This name has been chosen in analogy to the concept of characteristic functions in probability theory.

3.6.4.3 B-Spline Interpolation and Approximation as a Discrete-Time Filtering Process

Because the spline basis functions are compactly supported, interpolation or approximation based on a spline representation can be formulated as a discrete-time FIR filtering operation. Using the notation of basepoint sets, equation (3.165) is stated as a discrete-time convolution according to (3.96),[3]

$$y(t) = \sum_{k=0}^{N} c[n-k] \beta^N (\mu - t_k) \,. \tag{3.179}$$

The use of symmetric B-spline basis functions $\beta^N(t)$ implies a symmetric basepoint set (3.89). Consequently, the sample index n and the intersample position μ are calculated according to (3.85) and (3.86)

$$n = \text{round}\left(\frac{t}{T_i} + \frac{N}{2}\right) \qquad \text{and} \qquad \mu = \frac{t}{T_i} + \frac{N}{2} - n \,.$$

3.6.4.4 Cardinal Spline Interpolation

As observed from the general form of spline interpolation or approximation (3.165)

$$y(t) = \sum_{-\infty}^{\infty} c[n] \beta^N(t-n),$$

the first step in applying a spline interpolation or approximation method consists of determining the spline coefficients $c[n]$. To gain an interpolating spline curve that passes through the samples of the input sequence $x[n]$, the interpolation condition (3.33)

$$y(nT_i) = x[n] \quad \text{for } n \in \mathbb{Z}$$

must be enforced in the calculation of the coefficients $c[n]$. Methods that yield interpolating spline curves are denoted *cardinal spline interpolation* [Sch69, Sch73, Mei02]. This term has been chosen with reference to the cardinal interpolation function (3.31) proposed in [Whi15, Whi29]. Alternatively, the term *B-spline interpolation* is often used to denote interpolation methods based on splines, for instance [HA78, UAE91, Uns99]. However, in order to distinguish clearly between B-spline basis functions and interpolation methods based on these functions, the term "cardinal spline interpolation" is used exclusively in the following.

In numerical analysis, the spline coefficients $c[n]$ are calculated by solving a system of linear equations that enforces the interpolation condition for all data points as well as a suitable behavior at the end points of the interpolation range (e.g. [Pow91, BSMM06]). Due to its banded and symmetric matrix structure, efficient algorithms for solving this linear system exist. However, this approach is considered as ill-suited for the majority of signal processing applications. Most likely, this is due to the memory requirements proportional to the signal length and the inappropriateness for causal, real-time processing.

---

[3] Note that, in contrast to (3.96), the sampling period is normalized to $T_i = 1$ here.

In [GCB90, UAE91, UAE93a], an approach to calculate the spline coefficients by discrete-time filtering is developed

$$c[n] = (b^N)^{-1}[n] * x[n], \tag{3.180}$$

where $*$ denotes discrete convolution. The discrete sequence $(b^N)^{-1}[n]$ is the impulse response of a discrete-time filter that is formed by inverting the transfer function obtained from sampling the B-spline basis function $\beta^N(t)$ at the locations of the data points. For details, the reader is referred to the above references or [Uns99]. The transfer function corresponding to $(b^N)^{-1}[n]$ is rational and noncausal, but can be separated into a causal and an anticausal part.

In this way, the coefficients $c[n]$ for a finite sequence $x[n]$ are calculated by a recursive (IIR) filtering operation followed by a recursive filter that runs backward on the signal. Some additional calculations ensure sensible start values for the recursions. This algorithm is widely used in image processing and related fields.

However, for general signal processing applications that perform causal processing and operate on signals of conceptually infinite length, this approach is not applicable due to the anticausal recursive filtering required. For causal implementations or approximations of cardinal spline interpolation, references in literature are very scarce. Proposed solutions are the application of a FIR filter to compensate the magnitude error caused by the spline basis functions [ZB94] or linear-phase FIR approximations of the noncausal IIR prefilter [ESCA96, VS07].

While cardinal spline interpolation is efficiently implemented as a finite discrete-time convolution process (3.179) preceded by the calculation of the spline coefficients $c[n]$, this two-stage process complicates an analysis of the method. It is therefore advantageous to combine the two stages into a single continuous-time resampling function. This representation, denoted $\eta^N(t)$, is commonly referred to as cardinal spline representation [UAE93a] or cardinal spline basis functions [Uns99]. Using this function, cardinal spline interpolation is expressed by the convolution

$$y(t) = \sum_{n=-\infty}^{\infty} x[n]\eta^N(t), \tag{3.181}$$

where the basis functions of cardinal spline interpolation $\eta^N(t)$ are defined by

$$\eta^N(t) = \sum_{n=-\infty}^{\infty} (b^N)^{-1}[n]\beta^N(t-n). \tag{3.182}$$

In this way, $\eta(t)$ is formed as a continuous-time convolution of the spline basis function $\beta^N(t)$ and a pulse-modulated signal determined by the samples $(b^N)^{-1}[n]$.

In [Sch73], equation (3.181) is referred to as the cardinal Lagrange interpolation formula. The reference to Lagrange interpolation is due to the fact that the samples $x[n]$ are directly used in the interpolation formula alike in the Lagrange form of polynomial interpolation (3.134) (see [Sch46a]).

The impulse response $\eta^N(t)$ is depicted in Figure 3.34a for order $N = 3$. Since the prefilter $(B^N)^{-1}(z)$ is recursive and noncausal, $\eta(t)$ does not have compact support. In this way, $\eta(t)$ resembles the impulse response of the cardinal interpolation formula (3.30). However, the impulse response of cardinal spline interpolation converges to zero much more rapidly for
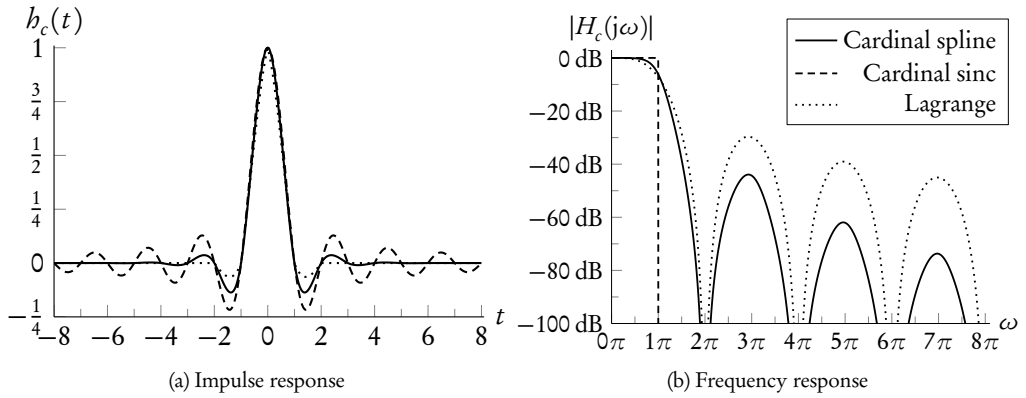
(a) Impulse response   (b) Frequency response

Figure 3.34: Continuous impulse and frequency response of cardinal spline interpolation of order $N = 3$.

increasing $|t|$ than the cardinal interpolation formula. In [Sch73], it is stated that the cardinal spline representation decays exponentially, that is, for each order N, positive constants A_N and α_N exist such that

$$|\eta(t)| < A_N e^{-\alpha_N|t|}. \tag{3.183}$$

Applied to practical implementations of cardinal spline interpolation, equation (3.183) implies that the coefficients of the discrete impulse response $(b^N)^{-1}[n]$ decrease rapidly as $|n|$ increases. Thus, truncation artifacts are less severe than in case of cardinal sinc interpolation if the impulse response is windowed or truncated. Additionally, the fast convergence of $(b^N)^{-1}[n]$ suggests that the prefilter $(B^N)^{-1}(z)$ can be approximated by a relatively short FIR filter without compromising the resampling quality.

On the other hand, the cardinal spline representation converges to the cardinal interpolation formula as the order N goes to infinity. This convergence is pointwise and holds for every L_p norm both in the time and the frequency domain [Sch74, AUE92].

The continuous frequency response of the cardinal spline interpolator of order $N = 3$ is shown in Figure 3.34b. In comparison to a Lagrange interpolator of the same order, cardinal spline interpolation exhibits superior image attenuation in the stopband as well as a less severe roll-off toward high frequencies in the passband.

3.6.4.5 Efficient Evaluation of B-Spline Basis Functions

Despite the various applications of splines, dedicated algorithms and implementation structures did not gain particular attention in the field of discrete-time signal processing. Typically, splines basis functions are regarded as piecewise polynomial resampling functions [BTU99, BTU01, VS07].

In the field of numerical analysis, the de Boor algorithm [dB72, Boo01b] and algorithms based on the explicit calculation of the B-spline basis functions [Cox78] are typically used.
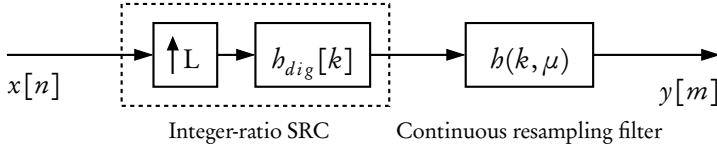
Figure 3.35: Implementation structure based on integer-ratio sample rate increase and a continuous resampling function.

While these methods show unconditional numerical stability [Cox72, Cox78], the ability to handle arbitrary sample positions (knots) with possibly coinciding knots plays a major role in the design and analysis of these algorithms. However, these features are not relevant for discrete-time signal processing on equidistant sequences. According to [Cox78], both algorithms require about $^3/_2N^2 + O(N)$ "long operations", which correspond to multiplications or divisions, and a proportional number of additions or subtractions. While several optimizations are applicable for equidistant sequences, such as the replacement of divisions by scaling factors, straightforward adaptations of these algorithms do not yield performance gains over polynomial-based interpolation filters.

For this reason, symmetric piecewise polynomial functions, which are realized by the modified Farrow structure, are considered as efficient implementations for symmetric B-spline basis functions in the following. Thus, the computational effort follows from Table 3.3 with the polynomial order set equal to the filter order, that is, $M = N$.

3.6.5 Implementation Structures Incorporating Integer-Ratio Oversampling

Structures incorporating an integer-ratio sample increase are widely used both in FD filtering and ASRC. The signal flow of a discrete-time implementation of this structure is shown in Figure 3.35. A system for integer-rate sample rate increase, consisting of a sample rate expander and a discrete-time anti-imaging filter $h_{dig}[k]$ as characterized in Section 3.4.2, is followed by a continuous-time resampling function implemented as a finite, discrete convolution with variable coefficients $h(k, \mu)$.

The integer-rate sample rate increase is typically implemented by means of efficient polyphase structures [BBC76, CR83], and the anti-imaging filter is usually designed as a lowpass filter according to the design specification (3.60). In most cases, $h(k, \mu)$ is a fixed resampling function with explicitly computable coefficients. Lagrange interpolation is often used in this role, mainly due to the good passband performance and the adherence to the interpolation condition. Moreover, this choice enables the use of the efficient implementation structures illustrated in Section 3.6.3 for the continuous-time resampling filter.

The widespread use of this structure has different reasons for FD filtering and ASRC. In fractional delay filtering, the main intention is to achieve good performance over a wide frequency range, which is costly to implement using single-stage structures based on fixed continuous-time resampling functions [MKK94, Her04, HJ05].

For arbitrary sample rate conversion, there are several reasons. First, ASRC evolved as an extension or generalization of rational SRC. In this way, structures incorporating oversampling

enable the large number of design methods and implementation structures for integer-ratio to be used. Second, the use of oversampling overcomes the limitations of the majority of fixed continuous resampling filters, namely poor passband performance for wideband signals and limited image attenuation [Hen02, VS07]. Conversely, in comparison with ASRC algorithms purely based on rational SRC, the use of sophisticated continuous-time resampling filters enables a significant reduction of the required oversampling ratio, coefficient memory and control effort [LPW82, Ram84].

3.6.6 The Generalized Farrow Structure

A modification to the Farrow structure that is referred to as the *Generalized Farrow structure* has been initially proposed in [Ram98]. It is similar to the Farrow structure, but features a set of L different coefficient sets for the subfilters $C_m(z)$ (3.109c). These coefficient sets are denoted as

$$C^{[l]} = \{c_{mn}^{[l]}\} \qquad \text{for } l = 0, 1, \ldots, L-1 \tag{3.184}$$

in the following. The generalized Farrow structure is depicted in Figure 3.36a.

Based on the intersample position μ, an integer index l, $0 \leq l < L$ and a transformed intersample position μ^* are computed

$$l = \lfloor (\mu - \mu_{min})L \rfloor \tag{3.185a}$$
$$\mu^* = (\mu - \mu_{min})L - l + \mu_{min}^*$$
$$= \text{frac}\left([\mu - \mu_{min}]L \right) + \mu_{min}^*. \tag{3.185b}$$

Here, the function $\text{frac}(x)$ denotes the fractional part of the argument x (e.g. [GKP06])

$$\text{frac}(x) = x - \lfloor x \rfloor. \tag{3.186}$$

Consequently, the transformed intersample position μ^* lies in the unit-sized intersample position range

$$\mu_{min}^* \leq \mu \leq \mu_{max}^* \qquad \text{with } \mu_{max}^* - \mu_{min}^* = 1. \tag{3.187}$$

The index l is used to select the coefficient set for the Farrow structure. So, an output value is calculated as

$$y(t) = \sum_{m=0}^{M} \left(\sum_{k=0}^{N} c_{mk}^{[l]} x[n-k] \right) (\mu^*)^m. \tag{3.188}$$

3.6.6.1 Relation to Multirate Structures

The derivation of the generalized Farrow structure was motivated by resampling filters based on integer-rate oversampling and continuous-time resampling functions [Ram98]. Indeed, the selection of a coefficient set can be conveniently described by a commutator model, a representation commonly used in rational SRC (see, e.g., Section 3.4.2.1). The resulting structure is shown in Figure 3.36b. It consists of L Farrow structures, represented by the transfer functions

$$H^{[l]}(z, \mu^*) = \sum_{m=0}^{M} \left(\sum_{n=0}^{N} c_{mn}^{[l]} z^{-n} \right) (\mu^*)^m \qquad \text{for } l = 0, 1, \ldots, L-1, \tag{3.189}$$

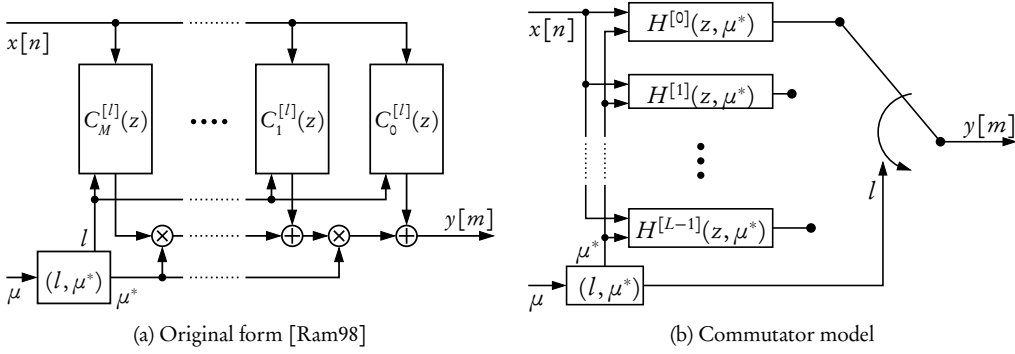(a) Original form [Ram98]   (b) Commutator model

Figure 3.36: Generalized Farrow structure.

each parameterized by a coefficient set $C^{[l]}$. These Farrow structures use the transformed intersample position μ^* as control variable. Depending on the index variable l, the output $y[m]$ is selected from one of these structures.

In [Hen02], the generalized Farrow structure is characterized as "storing more than one polyphase component" of the continuous impulse response characterizing the piecewise polynomial resampling filter. Conversely, the conventional Farrow structure is described as "computing all coefficients from one polyphase component".

3.6.6.2 Continuous-Time Impulse Response

As a modification of the Farrow structure, resampling based on the generalized Farrow structure can be expressed as a finite convolution (3.83)

$$y(t) = \sum_{k=0}^{N} x[n-k]h(k,\mu). \tag{3.190}$$

Because the generalized Farrow structure consists of L coefficient sets $C^{[l]}$ that are selected based on the intersample position, the discrete filter coefficients $h(n,\mu)$ are piecewise polynomial functions consisting of L polynomial pieces of order M. They can be expressed as

$$h(n,\mu) = h^{[l]}\left(n, \mathrm{frac}\left([\mu - \mu_{min}]L\right) + \mu^*_{min}\right) \quad \text{if } \mu_{min} + \frac{l}{L} \leq \mu \leq \mu_{min} + \frac{l+1}{L} \tag{3.191a}$$

where $h^{[l]}(n,\mu^*)$ denotes a single polynomial piece

$$h^{[l]}(n,\mu^*) = \sum_{m=0}^{M} c_{mn}^{[l]}(\mu^*)^m. \tag{3.191b}$$

Introducing basis functions $f^{[l]}(m,\mu)$ in the style of $f(m,\mu)$ (3.112)

$$f^{[l]}(m,\mu) = \begin{cases} \left(\mathrm{frac}\left([\mu - \mu_{min}]L\right) + \mu^*_{min}\right)^m, & \mu^*_{min} + \frac{l}{L} \leq \mu \leq \mu^*_{min} + \frac{l+1}{L} \\ 0, & \text{otherwise} \end{cases} , \tag{3.192a}$$
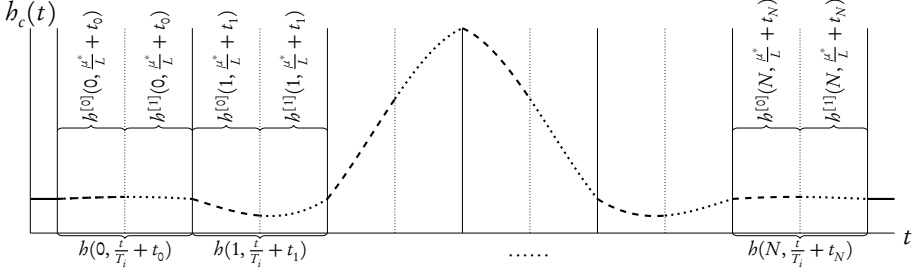
Figure 3.37: Continuous impulse response $h_c(t)$ of the generalized Farrow structure for $L = 2$.

the filter coefficients $h(n, \mu)$ can be expressed as

$$h(n, \mu) = \sum_{l=0}^{L-1} \sum_{m=0}^{M} c_{mn}^{[l]} f^{[l]}(m, \mu). \tag{3.192b}$$

Consequently, the continuous-time impulse response $h_c(t)$ is compactly represented by (3.113)

$$h_c(t) = \sum_{n=0}^{N} \sum_{l=0}^{L-1} \sum_{m=0}^{M} c_{mn}^{[l]} f^{[l]} \left(m, \tfrac{t}{T_i} + t_n \right). \tag{3.193}$$

In this way, $h_c(t)$ consists of the interleaved polynomial pieces of the L Farrow structures, each scaled to the length $\frac{T_i}{L}$ as shown in Figure 3.37.

It is worth noting that, even if $h_c(t)$ is symmetric with respect to zero, the polynomial pieces corresponding to a particular coefficient set $C^{[l]}$ do not exhibit such a symmetry in general. This behavior is related to polyphase implementations in rational SRC, where symmetry relations in the prototype filter are not retained in the polyphase branches which are obtained by de-interleaving the prototype.

3.6.7 The Transposed Farrow Structure

The implementation structures considered so far approximated the frequency response of an ideal resampling filter with a fixed cutoff frequency

$$\widehat{H}_c(\mathrm{j}\Omega) = \begin{cases} T_i, & |\Omega| < \frac{\Omega_i}{2} \\ 0, & |\Omega| \geq \frac{\Omega_i}{2} \end{cases}. \tag{3.194}$$

Consequently, these structures are not applicable to sample rate reductions, because in this case, the required cutoff frequency (3.54) depends on the output sampling rate to avoid baseband aliasing.

The transposed Farrow structure [HF00, Hen02, BVSR02] is an efficient implementation structure for sample rate reductions with variable conversion ratios $R \leq 1$.

Formally, the transformed Farrow structure is obtained by transposing the original Farrow structures using network theoretic transformations as described, for instance, in [CR83]. The
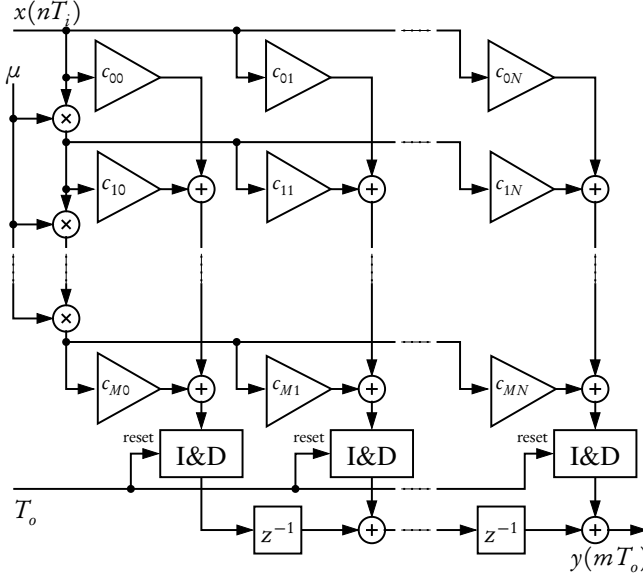
Figure 3.38: Transposed Farrow structure.

resulting structure is shown in Figure 3.38. In the transposed signal flow graph, the direction of all signals is reversed. Summations are replaced by branching operations and vice versa. The delay elements z^{-1} in the original Farrow structure are replaced by integrate-and-dump elements, denoted as "I&D". These elements accumulate the discrete-time input signals and are reinitialized to zero by the reset port. For the transposed Farrow structure, this reset is triggered by the output sampling clock T_o.

Internally, this operation corresponds to evaluating, that is sampling, the continuous-time impulse response $h_c(t)$ with the output sampling period T_o. In contrast, the conventional Farrow structure and its variants use the input period T_i to sample $h_c(t)$.

In this way, the characteristics of the anti-aliasing/anti-imaging filter $H_c(j\omega)$ are functions of the output sampling frequency, as required for variable-ratio sample rate decreases. In [HF00], the behavior of this structure is described as "the transfer zeros are clustered about the aliasing components at integer multiples of $\frac{1}{T_o}$ and thus, attenuate the aliasing components".

The transposed Farrow structure is not suited for sample rate increases. In this case, the structure does not provide sufficient image attenuation, because the cutoff frequency increases linearly with the output sampling frequency. Moreover, the computational effort to compute one output sample depends on the conversion ratio, because the number of samples of $h_c(t)$ evaluated for a single output value depends on T_o. As in case of the original Farrow structure, a generalized version consisting of multiple coefficient sets has been proposed [HF00, KG08].

It is worthwhile to note that the concept of sampling a continuous-time impulse response with the output sampling frequency is not unique to the transposed Farrow structure, but has been proposed much earlier. In [SG84], a sample rate conversion algorithm is described that is capable of both sample rate increases and decreases based on a single continuous-time impulse

response stored in parametric form.

While the impulse response is limited to a windowed sinc function approximated by a piecewise linear function, this algorithm features two distinct modes which are switched depending on the conversion ratio. In case of a sample rate reduction, the impulse response is sampled with the output sampling rate, resulting in the variable-cutoff anti-aliasing filter required for this case. In this way, this algorithm exhibits the features attributed to the transposed Farrow structure. At the same time, it enables sample rate increases by using the input sampling period to evaluate the continuous-time impulse response.

3.7 Design Methods for Variable Fractional Delay Filters

In the preceding section, numerous algorithms and implementation structures for interpolating or approximating the value of a discrete-time sequence at arbitrary output instants have been described. Most of these structures are applicable to variable fractional delay filtering. For the majority of these structures, a set of coefficients has to be determined by means of sophisticated design methods. These design methods are specific to VFD filtering. In this section, we provide a concise review of existing design methods. In contrast to other surveys on FD filtering, most notably [LVKL96], only the design of structures supporting variable delay values is considered here.

3.7.1 Design Methods for the Farrow Structure and its Variants

As described in Section 3.6.1, the Farrow structure, including modifications such as the modified Farrow structure, are variable digital filters. That is, the behavior of these filter structures is adjusted by a control variable. For FD filtering, the fractional delay d or a quantity derived from this value, typically by a linear transformation, is used as control variable.

The purpose of coefficient design for the Farrow structure is to determine the coefficients c_{mn} such that the frequency response $H(e^{j\omega}, d)$ approximates the frequency response of the ideal fractional delay element $\widehat{H}(e^{j\omega}, d)$ (3.38)

$$\widehat{H}(e^{j\omega}, d) = e^{-j\omega d} .$$

For the purpose of a uniform characterization as motivated in Section 3.5, the intersample position μ is used as control variable instead of d in the following. Because these quantities are uniquely related according to equation (3.95), the ideal frequency response as a function of the intersample position is represented by

$$\widehat{H}\left(e^{j\omega}, \mu\right) = e^{j\omega\mu} = \widehat{H}\left(e^{j\omega}, -d\right) . \tag{3.195}$$

It is noted that this conceptual representation does not include the implementation delay to enable a causal implementation, nor does it consider the role of different basepoint sets.

The approximation error of a VFD filter is a complex-valued function which is defined as the difference between the actual and the desired frequency response

$$E(\omega, \mu) = H\left(e^{-j\omega}, \mu\right) - \widehat{H}\left(e^{-j\omega}, \mu\right) . \tag{3.196}$$

In general form, the objective of optimal filter design is to minimize the norm of the approximation error over a given approximation region X

$$\underset{\{c_{mn}\}}{\text{minimize}}\, E_p = ||E(\omega, \mu)||_p\,, \tag{3.197}$$

where $||\cdot||_p$ denotes the L_p-norm (see, e.g. [Pow91])

$$||f(x)||_p = \left(\int\limits_X |f(x)|^p\, \mathrm{d}x \right)^{\frac{1}{p}}. \tag{3.198}$$

For filter design, the L_2 (least squares) norm $||\cdot||_2$ and the L_∞ (Chebyshev or minimax) norm $||\cdot||_\infty$ are most widely used

$$||f(x)||_2 = \sqrt{\int\limits_X |f(x)|^2\, \mathrm{d}x} \tag{3.199a}$$

$$||f(x)||_\infty = \max_X |f(x)|. \tag{3.199b}$$

For VFD design, the approximation region is two-dimensional, that is, a bounded subset of $\mathbb{R}^2$, because the frequency response (3.195) depends both on the normalized angular frequency ω and the intersample position μ

$$X = \{(\omega, \mu) \in X\,|\,0 \le \omega \le \omega_c \wedge \mu_{min} \le \mu \le \mu_{max}\}. \tag{3.200}$$

The cutoff frequency ω_c denotes the highest frequency of interest in the discrete-time signal.

3.7.1.1 Weighted Least Squares Design

Application of the L_2 norm, that is, setting $p = 2$ in (3.197), results in a least-squares approximation problem

$$\underset{\{c_{mn}\}}{\text{minimize}}\, E_2 = \int\limits_X \left| H\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) - \widehat{H}\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) \right|^2 \mathrm{d}\omega\, \mathrm{d}\mu. \tag{3.201}$$

Actually, this formula represents the minimization of the squared error norm, which is commonly referred to as the "integrated squared error" [Väl95a] or the "total squared error" [AS98]. However, the obtained solution also minimizes the L_2 norm because the square function is monotonically increasing for positive arguments. This design specification has been applied in the initial proposal of the Farrow structure [Far88].

A design method based on a weighted least squares criterion (WLS) has been proposed in [TCHR97]. WLS extends the least squares norm by weighting the error function (3.196) with a strictly nonnegative function $W(\omega, \mu)$, resulting in

$$\underset{\{c_{mn}\}}{\text{minimize}}\, E_{wls} = \int\limits_X \left(W(\omega, \mu) \left| H\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) - \widehat{H}\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) \right| \right)^2 \mathrm{d}\omega\, \mathrm{d}\mu \quad \text{with} \tag{3.202}$$

$$W(\omega, \mu) > 0 \qquad \text{for } (\omega, \mu) \in X\,.$$

In contrast to the original formulation, this representation includes the weighting function within the square term. This modification allows a uniform representation of the weighted error irrespective of the norm L_p used, as argued in [Sar93]. The design method in [TCHR97] discretizes both the frequency axis and the intersample position range into a dense grid of frequencies. In this way, the integral

$$E_{wls} = \int\limits_{\mu_{min}}^{\mu_{max}} \int\limits_{0}^{\omega_c} \left(W(\mu,\omega) \left| H\left(e^{-j\omega},\mu\right) - \widehat{H}\left(e^{-j\omega},\mu\right) \right| \right)^2 d\omega \, d\mu \qquad (3.203)$$

is replaced by a discrete summation

$$E_{wls} = \sum_{\mu \in C} \sum_{\omega \in F} \left(W(\mu,\omega) \left| H\left(e^{-j\omega},\mu\right) - \widehat{H}\left(e^{-j\omega},\mu\right) \right| \right)^2 \qquad (3.204)$$

with discretization grids C and F for the intersample position and the frequency variable, respectively. This intuitive discretization approach corresponds to numerical integration using the rectangle rule. However, it is valid only for equidistant grids, because the distance between grid points is not considered. The discretized weighting function $W(\omega,\mu)$ is specified as the product of two functions $W(\mu)$ and $W(\omega)$ that describe independent weights for the intersample position and frequency range, respectively.

$$W(\mu,\omega) = W(\mu)W(\omega) \qquad \text{for } \mu \in C \quad \text{and } \omega \in F. \qquad (3.205)$$

This type of weighting is referred to as separable weighting function and enables the least squares problem to be solved more efficiently. Although separable weighting functions limit the space of possible weightings, this kind of specification is commonly considered as sufficient for VFD filter design [LD99].

In the derivation of WLS methods for the Farrow structure, e.g. [TCHR97], relatively much work is devoted to transforming the two-dimensional, complex-valued approximation problem into the normal equation form of a least squares problem [BSMM06, Pow91], a matrix equation that can be directly solved as a system of linear equations. Furthermore, because the matrices involved in the approximation problem are prone to ill-conditioning, much effort is put into transforming the optimization problem into a form more suitable for numerical computation. However, ill-conditioning issues are intrinsic to weighted least squares filter design problems, since they result from the use of transition bands (don't care-regions) in the design specification (see e.g. [PB87, Sar93, Bur95, LVKL96]).

Further modifications to the WLS design method have been proposed in [LD99, Den01]. Here, the discretization of the frequency and intersample position axis is replaced by closed-form expressions of the involved integrals or, in one case, application of a numerical integration method. This algorithm also uses separable weighting functions $W(\mu)$ and $W(\omega)$, specified as piecewise linear functions of a continuous variable. The main advantage of this algorithm is a significant reduction of the computational effort for coefficient design, combined with slight improvements of the design error.

Typically, sophisticated adjustments of the weighting function $W(\mu,\omega)$ are limited to iterative improvements of the design error with respect to objectives different from pure least-squares

criteria. One common objective is to reduce the maximum error similar to a L_∞ design or a peak-constrained least-squares design [AS98]. In this way, WLS is often applied in the style of iterative weighted least squares design, e.g. [LLCY92, BBS94].

3.7.1.2 Weighted Least Squares Design for Symmetric Subfilters

As shown in Section 3.6.2, the use of linear-phase subfilters $C_m(z)$ with even or odd coefficient symmetries enables more efficient implementations such as the modified Farrow structure. For FD filtering, such structures have been proposed in [Tse04, Den04, DL06]. Aside from more efficient implementations, these symmetry conditions bear also advantages for the filter design process. The improvements are based on the symmetry of the frequency response of the ideal FD element for symmetric intersample position ranges

$$\widehat{H}\left(\mathrm{e}^{\mathrm{j}\omega}, -\mu\right) = \widehat{H}^*\left(\mathrm{e}^{\mathrm{j}\omega}, \mu\right). \tag{3.206}$$

Here, x^* denotes the complex conjugate of x. Imposing the coefficient symmetries (3.120) for the modified Farrow results in the same conjugate symmetry for the frequency response of the actual VFD filter

$$H\left(\mathrm{e}^{\mathrm{j}\omega}, -\mu\right) = H^*\left(\mathrm{e}^{\mathrm{j}\omega}, \mu\right). \tag{3.207}$$

Consequently, the system response is determined by one half of the intersample position interval, for instance $0 \le \mu \le \frac{1}{2}$. So, the design objective for WLS design can be stated as

$$E_{wls} = \int_0^{\frac{1}{2}} \int_0^{\omega_c} \left(W(\mu, \omega) \left| H\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) - \widehat{H}\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) \right| \right)^2 \mathrm{d}\omega \, \mathrm{d}\mu. \tag{3.208}$$

In this way, the design problem for the modified Farrow structure for FD filtering is less complex than for the conventional Farrow structure in two ways. First, the number of independent coefficients is approximately halved due to the imposed symmetries. Second, the size of the approximation region is reduced to 50 %. In case of discretized design methods, the number of required grid points is reduced by a factor of two while maintaining the same grid density. Both aspects reduce the size of the optimization problem and are likely to alleviate potential numerical conditioning issues.

3.7.1.3 Designs Based on a Weighted Chebyshev Norm

A method for design variable digital filters with respect to a weighted L_∞ norm has been proposed in [TYCT04]. The method is based on semidefinite programming, a method for constrained optimization that forms a subset of convex optimization [BV04]. Variable fractional delay filters are included as a design example. For weighted minimax design, the design specification is obtained from (3.197) by applying the L_∞ norm and incorporating a weighting function $W(\omega, \mu)$

$$\underset{\{c_{mn}\}}{\mathrm{minimize}} \, E_\infty = \max_{\substack{0 \le \omega \le \omega_c \\ \mu_{min} \le \mu \le \mu_{max}}} W(\omega, \mu) \left| H\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) - \widehat{H}\left(\mathrm{e}^{-\mathrm{j}\omega}, \mu\right) \right|. \tag{3.209}$$

For optimization, the passband $[0, \omega_c]$ and the intersample position range $[\mu_{min}, \mu_{max}]$ are discretized into grids $\{\omega_i\}$ and $\{\mu_i\}$, respectively. The discretized weighting function $W(\omega, \mu)$ is not required to be separable as in case of the WLS designs. In any case, the number of constraints in the optimization problem increases as the product of the grid sizes. Thus, the primary advantage of separable weighting functions in case of WLS designs cannot be brought to bear for this L_∞ design method.

The method proposed in [TYCT04] does not consider coefficient symmetries, since this condition is not applicable to the general class of variable digital filters targeted. The frequency responses $H\left(e^{j\omega}, \mu\right)$ and $\widehat{H}\left(e^{j\omega}, \mu\right)$ are modeled as variable, but causal FIR filters. As noted in the context of complex-valued minimax FIR filter design [ACL93, KM99], this design specification is prone to ill-conditioning. Therefore, it appears sensible to partition these filters into a common pure delay term $e^{-j\omega N/2}$ and a noncausal part in order to improve the conditioning of the design problem.

3.7.1.4 Designs Based on Digital Differentiators

A common approach to designing the subfilters $C_m(z)$ is to use approximations of discrete-time differentiators (e.g. [Har97, VHSR98, Tse02, PT03, JL03]). Expressing the frequency response of the ideal fractional delay filter

$$\widehat{H}\left(e^{j\omega}, \mu\right) = e^{j\omega \mu}$$

as a Maclaurin series, that is, a Taylor series evaluated at $\mu = 0$, results in

$$\widehat{H}\left(e^{j\omega}, \mu\right) = \sum_{k=0}^{\infty} \frac{1}{k!} (j\omega)^k \mu^k. \tag{3.210}$$

It is noted that, resulting from the use of the intersample position μ instead of the fractional delay value d, the coefficients of the expansion do not contain an alternating term $(-1)^k$ as typically found in derivations for FD filters, e.g. [Tse02, Fra08].

The frequency response of an ideal noncausal discrete-time differentiator of order k is given by [DRK93]

$$\widetilde{H}^{(k)}(e^{j\omega}) = (j\omega)^k. \tag{3.211}$$

Consequently, $\widehat{H}\left(e^{j\omega}, \mu\right)$ can be stated as

$$\widehat{H}(e^{j\omega}, \mu) = \sum_{k=0}^{\infty} \frac{1}{k!} \widetilde{H}^{(k)}(e^{j\omega}) \mu^k. \tag{3.212}$$

In this way, the ideal fractional delay element is expressed as a sum of discrete-time differentiators weighted by monomials of μ. Truncation of the series (3.212) thus results in the frequency response of a Farrow structure. In [JL03], error bounds for the complex error, the phase error, and the phase delay error of the Farrow structure based on this truncated Taylor expansion are derived.

However, the relation between the Taylor series expansion and ideal differentiators does not imply that the subfilters of the Farrow structure are required to approximate discrete-time differentiators. Conversely, it has been shown in [DN04, DL06] that the rate of convergence of

the Taylor series to the ideal fractional delay is relatively slow. Thus, the VFD filters gained from this design method are not necessarily optimal with respect to a given error measure. Farrow structures for Lagrange interpolation, which are optimal with respect to the criterion of maximal flatness form a notable exception. In this case, it has been shown [SAS04, Den07a, Fra08] that the subfilters $C_m(z)$ represent maximally flat differentiators of order m scaled by $1/m!$.

3.7.1.5 Filter Designs Incorporating Phase Delay Specifications

A design method that incorporates phase delay specifications has been developed in [VS96b, VS97, VS00]. Although adapted to FD filtering, this design is based on the original derivation of the modified Farrow structure [VS96a] (Section 3.6.2.1), utilizing a symmetric piecewise polynomial continuous impulse response $h_c(t)$.

The design specification is given by

$$\underset{\{c_{mn}\}}{\text{minimize}} \; \delta_p = \underset{\substack{0 \le \omega \le \omega_c \\ \mu_{min} \le \mu \le \mu_{max}}}{\max} \; \left| \tau_p(\omega,\mu) - \widehat{\tau}_p(\omega,\mu) \right| \tag{3.213a}$$

$$\text{subject to} \quad \underset{\substack{0 \le \omega \le \omega_c \\ \mu_{min} \le \mu \le \mu_{max}}}{\max} \; \left\| H((\omega,\mu)| - 1 \right| \le \delta_{worst}. \tag{3.213b}$$

That is, the objective is to minimize the maximum error of the phase delay $\tau_p(\omega,\mu)$ while keeping the amplitude error within a prescribed limit δ_{worst}. Here, $\tau_p(\omega,\mu)$ denotes the phase delay of the noncausal interpolation filter as a function of the intersample position. The ideal phase delay follows from (3.46a)

$$\widehat{\tau}_p(\omega,\mu) = -\frac{\sphericalangle \widehat{H}(\omega,\mu)}{\omega} = -\mu, \tag{3.214}$$

where $\widehat{H}(\omega,\mu)$ denotes the noncausal frequency response of the ideal FD element.

As an extension of this problem, the interpolation condition (3.33) is included as an optional constraint. That is, the frequency is restricted to unity for intersample positions μ that correspond to locations of the samples $x[n]$. For N odd, this condition must be fulfilled for $\mu = \pm\frac{1}{2}$. This constraint binds some degrees of freedom for the filter coefficients c_{mn}. The remaining degrees of freedom are utilized to minimize the objective function (3.213). Because this problem involves the phase delay and thus the unwrapped phase of the complex argument $H(\omega,\mu)$, neither linearity nor convexity properties hold.

In [VS96b, VS97], this problem is solved using an algorithm for constrained minimax optimization [DV77].

As a notable result, this approach states a lower limit for the achievable maximum amplitude error $|H(e^{j\omega},\mu)| - |\widehat{H}(e^{j\omega},\mu)|$ for a frequency range $[0,\omega_c]$. This quantity is used to specify an appropriate error bound δ_{worst} for the constraint (3.213b). Since the formulation of the modified Farrow structure used in [VS96b] is restricted to odd subfilter orders N, the worst case amplitude error occurs for a half-sample delay, i.e. $\mu = 0$, corresponding to $\mu' = \frac{1}{2}$ in the formulation (3.115). In this way, δ_{worst} is obtained by designing a linear-phase filter of order N that approximates a half-sample delay in the frequency range $[0,\omega_c]$.

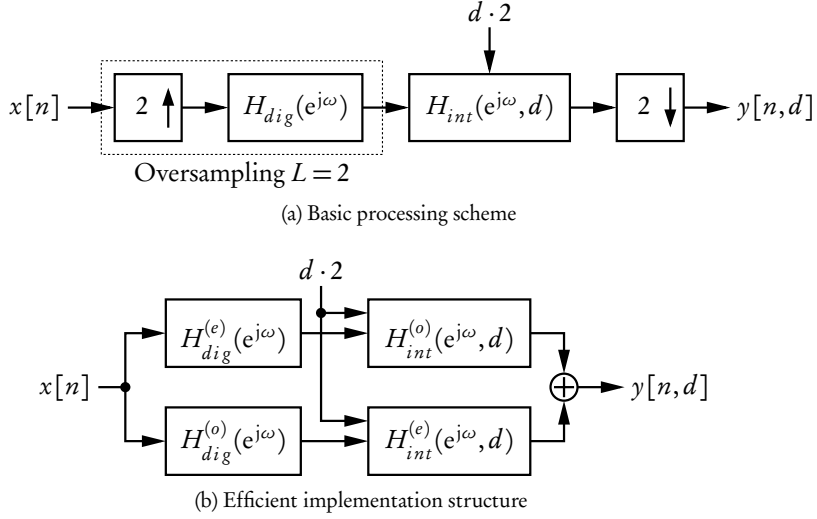(a) Basic processing scheme

(b) Efficient implementation structure

Figure 3.39: Wideband fractional delay element according to [MKK94].

3.7.2 Design for FD Filters Incorporating Integer-Ratio Oversampling

In the context of FD filtering, the use of structures involving oversampling apparently has been first proposed in [MKK94]. The basic processing scheme of this structure is depicted in Figure 3.39a. In the proposed filter structure, the oversampling ratio L is fixed to 2. The primary purpose of this structure is described as to increase the usable bandwidth of the filter without an excessive increase in the order of the variable fractional delay filter $H_{int}(e^{j\omega}, d)$, which is implemented as a Lagrange interpolator. According to [MKK94], the performance gain results from the relaxed requirements on the Lagrange interpolation filter. This interpolator has to comply with the design specifications only for half the bandwidth of the input signal. The anti-imaging filter $H_{dig}(e^{j\omega})$ is designed as a halfband filter using a windowing method.

An efficient implementation structure for this system is shown in Figure 3.39b. This structure operates at the nominal sampling frequency of the FD filter, that is, it is implemented as a single-rate structure. The decomposition of the filters $H_{dig}(e^{j\omega})$ and $H_{int}(e^{j\omega}, d)$ into filters containing the even and odd coefficients represents polyphase decompositions as described in Section 3.4.2. The resulting polyphase filters are denoted by the superscripts (e) and (o), respectively. This transformation eliminates the need for explicit sample rate expander or compressor functionalities, and the output sequence is formed by summing the outputs of the two polyphase branches.

It is noted that the multiplication by 2 at the output of the structure that is represented by a separate processing stage in [MKK94, Her04, HJ05] is not included here. As argued in [CR83], the required gain $L = 2$ is part of the filter specification for the anti-imaging filter $H_{dig}(e^{j\omega})$ of a sample rate increase and is trivially integrated into this filter.

The performance of the two-stage approach is further analyzed in [Her04], stating a reduced computational effort if a halfband filter is used as anti-imaging filter $H_{dig}(e^{j\omega})$. However, the

approach described there focuses on Lagrange interpolation filters using the Farrow structure. Thus, the results are not comparable to approaches that use efficient implementations for Lagrange interpolation as characterized in Section 3.6.3. In [HJ05], a technique is proposed to jointly optimize the filters $H_{dig}(e^{j\omega})$ and the subfilters of a Farrow structure implementing $H_{int}(e^{j\omega}, d)$. Conceptually, the obtained structure is identical to the generalized Farrow structure (see Section 3.6.6), albeit applied to FD filtering and limited to an oversampling ratio $L = 2$. Moreover, an iterative optimization is proposed to optimize the length of the subfilters $C_m(z)$ separately, similar to the procedure described in [JL03]. The computational effort of this method is characterized as superior to both single-stage FD filters based on the Farrow structures and the two-stage approach based on separately optimized subfilters. However, the optimization procedure is not described in sufficient detail to reproduce these results.

3.8 Design Methods for Arbitrary Sample Rate Conversion

Although the implementation structures for ASRC are identical to those used for VFD filters in most cases, the associated design methods are profoundly distinct. The most essential reason is that VFD filters are purely discrete-time structures, while ASRC filters for arbitrary ratios are conveniently modeled as a continuous-time system. A second distinction arises from the historical development of ASRC, which partly evolved as an extension of rational-ratio SRC, and is partly based on interpolation or approximation techniques not initially developed in the field of DSP. For this reason, systematic design methods, which yield optimal parameterizations for a given specification, are less common than in other branches of DSP, for instance filter design. This applies in particular to systems that are composed of different components, e.g. cascades of rational-factor SRC and continuous-time resampling functions.

   This section introduces design specifications and reviews existing approaches to coefficient design for ASRC implementation structures.

3.8.1 Design Objectives

Because ASRC is appropriately modeled by the hybrid analog/digital model, the design objectives are consequently expressed in the continuous-time domain or in the continuous Fourier transform domain. The ideal frequency of the continuous-time anti-imaging/anti-aliasing filter (3.53) of the hybrid analog/digital model

$$\widehat{H}_c(j\Omega) = \begin{cases} T_i, & |\Omega| < \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \\ 0, & |\Omega| \geq \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \end{cases} \tag{3.215a}$$

$$= \begin{cases} T_i, & |\Omega| < \Omega_c \\ 0, & |\Omega| \geq \Omega_c \end{cases} \quad \text{with } \Omega_c = \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \tag{3.215b}$$

has been introduced in Section 3.4.5. As argued there, systems for increasing and for decreasing the sampling rate differ qualitatively. In this section, only design methods for systems that do not decrease the sampling rate are considered. This restriction is in accordance with the majority of texts considering coefficient design for ASRC, e.g. [Ves99, Eva00b, VS07] and is motivated by the following reasons:

First, in the sample rate reduction case, the desired frequency response becomes a function of the cutoff frequency (3.215b). Consequently, the design is dependent on the conversion ratio R for most implementation structures, which contradicts the desirable property of ASRC systems to support variable conversion ratios.

Second, for most implementation structures, the designs obtained for sample rate increases can also be utilized for sample rate decreases by using the corresponding transposed structures, for instance the transposed Farrow structure (Section 3.6.7).

Finally, as will be shown in the following chapter, structures for sample rate reductions are not suitable for WFS signal processing.

With the restriction to nondecreasing conversion ratios, the ideal continuous frequency response simplifies to

$$\widehat{H}_c(j\Omega) = \begin{cases} T_i, & |\Omega| < \frac{\Omega_i}{2} \\ 0, & |\Omega| \geq \frac{\Omega_i}{2} \end{cases}. \tag{3.216}$$

Using the angular frequency ω normalized with respect to the input sampling period

$$\omega = \frac{2\pi f}{f_i} = 2\pi f T_i = \Omega T_i, \tag{3.217}$$

the ideal frequency response (3.216) can be stated as

$$\widehat{H}_c(j\omega) = \begin{cases} T_i, & |\omega| < \pi \\ 0, & |\omega| \geq \pi \end{cases}. \tag{3.218}$$

The passband amplitude of T_i stems from the frequency response of the ideal reconstruction filter (3.25). For the practical design of ASRC algorithms, this scaling has no impact. In accordance with the reasons stated in [VS07], the normalization $T_i = 1$ is used in actual design methods as well as in the presented frequency responses.

As explained in Section 3.5.4, resampling filters with real-valued, symmetric impulse responses exhibit convincing advantages. For this reason, they are used almost exclusively in ASRC. Consequently, the continuous frequency response is also a real function that is symmetric with respect to $\omega = 0$. Therefore, it suffices to approximate the desired frequency response over the region of positive frequencies $\omega \in [0, \infty)$ only.

The ideal frequency response (3.218) is not realizable by a causal filter. Moreover, a direct approximation of this specification by a finite-length filter typically leads to unacceptable errors in the frequency response, resulting from Gibbs phenomenon. For this reason, alternative design objectives are specified, which typically introduce transition regions or don't-care bands, that is, frequency intervals in which the frequency response is of no interest. In this way, the positive frequency axis is partitioned into disjoint regions: the passband X_p, the stopband region X_s, and one or multiple transition bands, which are denoted by ϕ. Consequently, the desired frequency response can be stated as

$$\widehat{H}_c(j\omega) = \begin{cases} 1, & |\omega| \in X_p \\ 0, & |\omega| \in X_s \end{cases}. \tag{3.219}$$

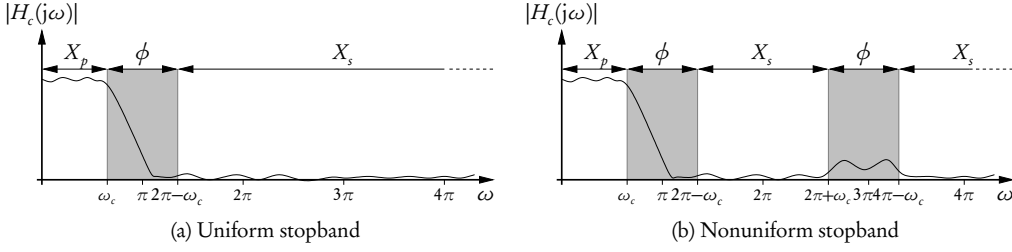(a) Uniform stopband   (b) Nonuniform stopband

Figure 3.40: Approximation region specifications for ASRC design. X_p is the passband, X_s denotes stopbands and ϕ are the transition bands.

which specifies the desired frequency response in the approximation region

$$X = X_p \cup X_s. \tag{3.220}$$

For SRC applications, these regions are determined by the highest frequency of interest in the input signal, ω_c [CR83], resulting in a passband edge ω_p and stopband a edge ω_s

$$\omega_p = \omega_c \tag{3.221a}$$

$$\omega_s = 2\pi - \omega_c. \tag{3.221b}$$

Using these edge frequencies, the passband region X_p is given by

$$X_p = \left[0, \omega_p\right]. \tag{3.222}$$

Likewise, the primary transition region ϕ between passband and stopband edge is the open interval

$$\phi = \left(\omega_p, \omega_s\right). \tag{3.223}$$

Since the continuous Fourier transform of the input signal $X(\mathrm{j}\omega)$ consists of periodic replications of the discrete-time spectrum $X(\mathrm{e}^{\mathrm{j}\omega})$, the don't-care bands of the input signal are also replicated periodically. For this reason, two different specifications for the stopband region X_s exist: uniform and nonuniform stopbands. A uniform stopband specification consists of a single stopband that spans the complete frequency interval of above ω_p, corresponding to the specification of desired frequency response

$$\widehat{H}_c(\mathrm{j}\omega) = \begin{cases} 1, & |\omega| \leq \omega_p \\ 0, & |\omega| \geq \omega_s \end{cases}. \tag{3.224}$$

In contrast, in a nonuniform stopband specification, the spectral replications of the don't-care region of the input signal $(\omega_p, \pi]$ are also included in the transition region ϕ, resulting in

$$\widehat{H}_c(\mathrm{j}\omega) = \begin{cases} 1, & |\omega| \leq \omega_p \\ 0, & 2k\pi - \omega_p \leq |\omega| \leq 2k\pi + \omega_p \quad \text{for } k = \pm 1, \pm 2, \dots \end{cases}. \tag{3.225}$$

Design specification	Passband X_p	Stopband X_s	Transition regions ϕ
Ideal lowpass	$[0, \pi/2)$	$[\pi/2, \infty)$	—
Uniform stopband	$[0, \omega_c]$	$[2\pi - \omega_c, \infty)$	$(\omega_c, 2\pi - \omega_c)$
Nonuniform stopband	$[0, \omega_c]$	$\bigcup_{k=1}^{\infty}[2k\pi - \omega_c, 2k\pi + \omega_c]$	$\bigcup_{k=1}^{\infty}(2[k-1]\pi + \omega_c, 2k\pi - \omega_c)$

Table 3.5: Passband, stopband and transition regions for design specifications for ASRC. The operator $\cup$ denotes a set union.

These approximation regions are schematically depicted in Figure 3.40, and the corresponding passband, stopband and transition regions are summarized in Table 3.5.

The choice of the stopband specification ultimately depends on the application: If the energy of the input signal in the don't-care region $(\omega_c, \pi]$, is negligible, then a nonuniform stopband specification (3.225) is sufficient. Otherwise, a uniform stopband (3.224) is required to attenuate the spectral replications of these signal components that would alias into the discrete-time output signal [Eva00b, VS07]. However, the potential performance gain from utilizing additional transition bands is very limited in most cases, except for narrowband input spectra. This behavior is in accordance with the observations reported in the field of rational SRC [CR83]. However, using the same reasoning as in case of rational SRC, the frequency response $H_c(j\omega)$ must provide sufficient attenuation within the transition regions to avoid artifacts due to the amplification of noise, for instance introduced in preceding processing stages.

Error measures   As in case of VFD filters and in general filter design problems, the L_2 norm (3.199a) and the L_∞ (3.199b) norm are the most widely used norms for optimal design of resampling filters. Typically, a strictly positive weighting function $W(\omega)$ is included in the design specifications

$$E_2 = \int_X \left(W(\omega) \left| H_c(j\omega) - \widehat{H}_c(j\omega) \right| \right)^2 d\omega \qquad (3.226a)$$

$$E_\infty = \max_{\omega \in X} W(\omega) \left| H_c(j\omega) - \widehat{H}_c(j\omega) \right|, \qquad (3.226b)$$

where X denotes the approximation region according to Table 3.5. Strictly speaking, equation (3.226a) represents the weighted squared error as discussed in Section 3.7, and minimization of E_2 results in a weighted least squares design. For L_2 designs, the weighting function is typically adjusted to control the error in specific intervals of the approximation region, for instance to limit the peak deviation.

In contrast, for L_∞ designs, $W(\omega)$ is primarily used to achieve a desired ratio between the maximum passband and stopband error (e.g. [VS07]). As in case of general discrete-time filter design, approximation with respect to the L_∞ norm is conveniently described by a tolerance scheme as depicted in Figure 3.41 that defines the maximum permissible passband and stopband
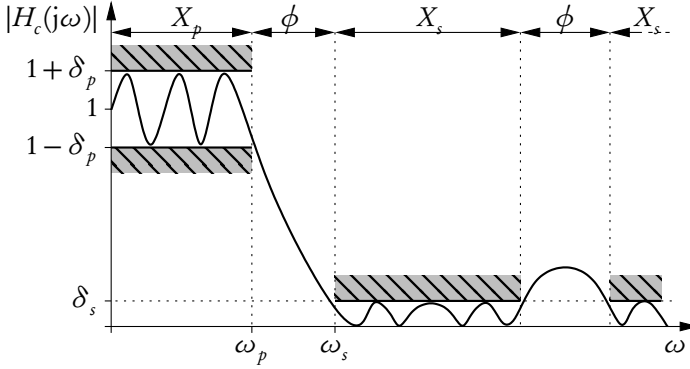
Figure 3.41: Approximation scheme for ASRC filters using passband and stopband toler-
        ances.

errors δ_p and δ_s

$$\delta_p = \max_{\omega \in X_p} |H_c(j\omega) - \widehat{H}_c(j\omega)| \tag{3.227a}$$

$$\delta_s = \max_{\omega \in X_s} |H_c(j\omega) - \widehat{H}_c(j\omega)|. \tag{3.227b}$$

Other measures directly related to the maximum passband and stopband error are the maximum
passband variance and the minimum stopband attenuation [Sar93]

$$A_p = 20 \log_{10} \left(\frac{1 + \delta_p}{1 - \delta_p} \right) \qquad \text{[in dB]} \tag{3.228a}$$

$$A_s = -20 \log_{10} (\delta_s) \qquad \text{[in dB]}. \tag{3.228b}$$

To achieve a desired ratio between the maximum passband and stopband error in L_∞ designs,
a weighting function is typically used which assumes constant values $W(X_p)$ and $W(\tilde{X}_s)$ in the
passband and the stopband regions

$$\frac{W(X_p)}{W(X_s)} = \frac{\delta_s}{\delta_p}. \tag{3.229}$$

This specification is identical to the use of weighting functions in the minimax design of discrete-
time FIR filters [PM72a, Sar93, OSB99].

It is important to emphasize that in ASRC design, the approximation region X is conceptually
of infinite extent. In contrast, the frequency response of a discrete-time filter is periodic with
respect to the frequency variable. Consequently, discrete-time filter design methods operate on a
single period of the frequency response only. This distinction influences both design methods
and the analysis of ASRC filters.

Numerical approximation techniques are generally based on functions defined on fixed inter-
vals [Pow91], regardless whether a discretization is performed. For this reason, ASRC design

methods based on approximating the desired frequency response must introduce a finite upper limit ω_{up} for the approximation region. Thus a finite uniform stopband region is given by

$$X_s = [2\pi - \omega_c, \omega_{up}]. \tag{3.230}$$

Likewise, the nonuniform stopband region specified in Table 3.5 is altered to

$$X_s = \left(\bigcup_{k=1}^{\infty} [2k\pi - \omega_c, 2k\pi + \omega_c] \right) \cap [2\pi - \omega_c, \omega_{up}], \tag{3.231}$$

where $\cap$ denotes set intersection.

However, the basis functions underlying all implementation structures for ASRC considered in this work exhibit an attenuation towards higher frequencies. Consequently, an upper limit for the approximation region can be generally found such that the quality of the designed filter is not degraded by the choice of ω_{up}. This upper frequency limit depends on the method in use and its design parameters, and it has to be determined heuristically. Thus, a validation of the obtained filter coefficients and possible iterations of the design process is inevitable in general.

The other problem comprises the error norms used to assess the quality of a design, which are also defined on the unbounded approximation region X. For the L_∞ norm, the observed damping of the basis functions ensures that the maximum error computed over a subset of X with a sufficiently large upper limit ω_{up} is identical to the maximum error over the infinite region X. For the L_2 norm, the numerically obtained norm depends on the chosen upper limit ω_{up}. Therefore, it is necessary to state the upper frequency limit that is used in the numerical calculation of the L_2 error. However, similar to the L_∞ norm, the error due to a finite upper integration boundary is negligible in most cases if ω_{up} is chosen sufficiently large, because of the attenuation of the basis functions of $H_c(\mathrm{j}\omega)$ toward high frequencies.

Additional Frequency- or Time-Domain Conditions   In addition to the frequency-domain-oriented error norms, it is often beneficial to specify additional conditions or constraints to be included in the design specification. In the literature on the design of discrete-time FIR filters, several constraints on the frequency response, such as peak error constraints [SLB96, Eva00b] or flatness constraints [Her71, Vai84] have been proposed. As these conditions can be typically stated as linear inequalities or equalities, they can be incorporated into most design methods based on optimization frameworks such as linear programming [Rab72, Sar93] or convex optimization (for an extensive survey, see e.g. [Dav10]).

Frequency-domain conditions can be used to control the filter magnitude at particular points or intervals of the pass- or stopband or to limit the amplification in the transition regions.

For resampling filters, a number of sophisticated conditions based on properties of the continuous impulse response $h_c(t)$ have been proposed [Eva00b, VS07], including the following:

Interpolation condition   The interpolation condition, introduced in Section 3.2.2, denotes the property of the resampling filter to reproduce the signal at input sampling locations exactly. In terms of the continuous impulse response, this condition is represented by

$$h_c(t) = \begin{cases} 1, & t = 0 \\ 0, & t = kT_i \end{cases} \qquad \text{for } k = \pm 1, \pm 2, \dots . \tag{3.232}$$

Continuity The condition of continuity indicates whether the continuous-time impulse response $h_c(t)$ is a continuous function. This can be formally stated as $h_c(t) \in C^0$, where C^0 denotes the space of continuous functions (see Section 3.6.4.2). Since $h_c(t)$ is typically a piecewise function composed of continuous basis function, this condition requires the adjacent pieces of $h_c(t)$ to attain identical values at the junction points. This condition can be often stated as a set of linear equations.

Differentiability As an extension to continuity, also denoted as regularity (e.g. [Mei02,BTU03]), this condition requires $h_c(t)$ to be k times continuously differentiable. In this way, $h_c(t)$ belongs to the differentiability class C^k. For $k = 0$, this condition is identical to continuity.

The benefit of these conditions generally depends on the application. However, the impact of these constraints on quality measures relevant to signal processing appears to be rather limited. For instance, the number of continuous derivatives is considered an important property in the field of mathematical interpolation formulas [Gre44, Sch46a]. Consequently, this condition has often been adopted in signal processing, e.g. [Key81]. However, in [BTU01, BTU03], it is conjectured that there is no stringent interrelation between regularity and quality for approximation applications, that is, the kind of applications most relevant to signal processing.

3.8.2 Design for the Modified Farrow Structure

As discussed in Section 3.6, the Farrow structure and its several variants are an efficient means to implement resampling filters based on finite, piecewise polynomial impulse responses. For this reason, ASRC systems based on these structures are often termed *polynomial-based interpolation filters* [VS96a, Ves99, VS07]. The restriction to real-valued, symmetric impulse responses, which has been motivated in Section 3.5.4, results in coefficient symmetries according to (3.120) if the intersample position range is chosen appropriately. Consequently, ASRC systems based on piecewise polynomial impulse responses can be generally implemented by the modified Farrow structure, thus combining the advantageous impulse response symmetries with the increased computational efficiency offered by this structure.

The general design objective for ASRC structures based on piecewise polynomial resampling filters is to determine the coefficient b_{mn} of the coefficient matrix B (3.129) parameterizing the modified Farrow structure. In general, the design methods fall into three distinct classes: Designs based on numerical interpolation or approximation methods, and optimization with respect to either time- or frequency-domain criteria.

3.8.2.1 Numerical Interpolation or Approximation Methods

The majority of numerical methods to interpolate or approximate a continuous function from a discrete set of data points is based on the evaluation of polynomials (see e.g. [Mei02]). Notable examples include all forms of polynomial interpolation (e.g. Lagrange, Hermite or Newton interpolation), osculatory interpolation [Gre44, Sch46a], spline basis functions [Sch46a, Uns99], or specialized kernels such as the interpolation formula proposed by Keys [Key81]. Moreover, they typically exhibit symmetric impulse responses. Consequently, all these methods can be implemented by the modified Farrow structure, and the polynomial coefficients b_{mn} are generally obtained without conceptual difficulties.

Due to their origin in the approximation or interpolation of functions, these methods are often referred to as time-domain interpolation methods (e.g. [Ves99, VS07]), as they are considered to be designed with respect to the domain of the original independent variable only. However, this is not entirely true, because many numerical interpolation formulas, for instance Lagrange or spline interpolation, have well-defined frequency-domain characteristics. For Lagrange interpolators, these properties will be discussed in detail in Section 5.3. Notwithstanding, the frequency-domain performance is most often inferior to that of a Farrow structure specifically designed to maximize this objective.

Though all methods mentioned here can be readily implemented by the modified Farrow structure, this is not necessarily the most efficient means. For instance, as shown in Section 3.6.3, sophisticated structures for Lagrange interpolation outperform implementations based on the modified Farrow structure even for low interpolation orders.

In any case, the use of numerical interpolation or approximation formulas does not require sophisticated coefficient design beyond the selection of an appropriate method and the determination the interpolation order.

3.8.2.2 Time-Domain Approximation

Another approach to coefficient design for the modified Farrow structure operates directly in the time domain. The design objective is to approximate the desired continuous-time impulse response $\widehat{h}_c(t)$ of a resampling filter by a realizable piecewise polynomial function $h_c(t)$ such that the approximation error becomes minimal with respect to an appropriate norm. This kind of design methods is described in detail in [Eva00a, Eva00b]. These designs are characterized by a set of different choices to be made.

The first distinction is concerned with the desired impulse response $\widehat{h}_c(t)$ to be approximated. In the most direct approach, the cardinal sinc interpolation formula (3.28), corresponding to a resampling filter specification with no transition region, is utilized. However, approximating this function by a piecewise polynomial of finite support causes severe truncation errors due to the slow damping of the sinc function. These errors result in frequency-domain magnitude deviations due to the Gibbs phenomenon in the same way as in case of FD filtering (see e.g. [CMT94, LVKL96, Väl95a]).

One way to alleviate this problem is to taper $\widehat{h}_c(t)$ by a window function. The use of a Kaiser window is proposed in [SG84, Smi11], albeit this approach uses multiple polynomial segments for each filter coefficient $h(k, \mu)$, thus resembling the generalized Farrow structure (Section 3.6.6).

In [Eva00a, Eva00b], the desired impulse response is specified as the ideal impulse response of a lowpass filter with a spline transition band [BSG92, Bur95]

$$\widehat{h}_c^{(spline)}(t) = A \operatorname{sinc}(\omega_0 t) \operatorname{sinc}\left(\frac{\Delta\omega t}{2p}\right)^p \quad \text{with } \omega_0 = \frac{\omega_p + \omega_s}{2} \text{ and } \Delta\omega = \omega_s - \omega_p, \quad (3.233)$$

where ω_p and ω_s denote the passband and stopband edges of the desired frequency response. A is a constant amplitude scaling factor and p denotes the spline order of the transition band specification. In this way, the bandwidth of approximation is reduced. At the same time, the impulse response becomes a smooth function, thus considerably reducing Gibbs phenomenon.

The desired impulse response $\widehat{h}_c^{(spline)}(t)$ inherently fulfills the interpolation condition (3.32). In the context of FD filtering, an equivalent approach utilizing least-squares approximation with smooth transition bands is described in [LVKL96].

The approximating function $h_c(t)$ forms a second choice. In [Eva00a, Eva00b], either single polynomials or piecewise polynomials are considered. However, the use of a single polynomial only reduces the number of coefficients to be stored, but requires the same number of operations to evaluate the filter coefficients $h(k, \mu)$ compared to a piecewise polynomial representation. On the other hand, resampling filters based on a single polynomial show inferior approximation quality, because the corresponding filter design problem offers significantly less degrees of freedom compared to an impulse response based on piecewise polynomials.

The objective function for the approximation constitutes a third design choice. In [Eva00b], the L_2 and the L_∞ norm are considered, and the interpolation condition (3.32) is incorporated as an additional, optional design constraint. The error analysis performed therein describes a number of severe limitations of this time-domain approach. For L_2 designs, the error tends to be dominated by the truncation error as the order of the piecewise polynomials increases. Consequently, a lower limit for the L_2 error dependent on the impulse response length is conjectured. Thus, it is necessary to find a sensible compromise between the truncation error and the approximation error within the finite support of $h_c(t)$. Such a trade-off can be achieved by an appropriate choice for the desired impulse response $\widehat{h}_c(t)$. If the desired frequency response representation based on spline transition bands (3.233) is used, then a viable solution is found by adjusting the spline order p.

An optimization procedure with respect to the L_∞ norm has been proposed by the same author [Eva00b]. It exhibits basically the same properties as the L_2 designs, but is prone to numerical instability due to ill-conditioning of the involved matrices.

3.8.2.3 Frequency-Domain Approximation

Designing the coefficients for the modified Farrow structures in the frequency domain offers several advantages compared to methods based on the time domain. First, in DSP applications, the frequency-domain characteristics of a system are typically more meaningful than pure time-domain criteria. Second, as the primary artifacts of ASRC systems, namely aliasing and imaging, are best described by the spectral properties of the resampler, a frequency-domain approximation enables a purposeful control of these errors. In any case, the frequency-domain description does not preclude time-domain specifications, because such conditions are typically handled by appropriate design methods.

A frequency-domain description for polynomial-based resampling filters and corresponding design methods based have been proposed in [Ves99, VS07]. This derivation is based on the original formulation of the modified Farrow structure (3.117) described in Section 3.6.2.1. This description uses a particular form of the convolution sum and a transformation of the intersample position range $0 \le \mu < 1$. Moreover, it is restricted to odd subfilter orders N.

Utilizing the symmetry conditions for the modified Farrow structure (3.118), the impulse response can be stated in an alternate form using modified basis functions $g'(m, n, t)$

$$h_c(t) = \sum_{m=0}^{M} \sum_{n=0}^{\frac{N-1}{2}} c'_{mn} g'(m, n, t) \quad \text{with} \tag{3.234a}$$

$$g'(m,n,t) = (-1)^m f'\left(m, \frac{t}{T_i} + n + 1\right) + f'\left(m, \frac{t}{T_i} - n\right) \tag{3.234b}$$

$$= \begin{cases} (-1)^m \left(\left[2\frac{t}{T_i} + n\right] - 1\right)^m, & -(n+1)T_i \leq t < -nT_i \\ \left(\left[2\frac{t}{T_i} - n\right] - 1\right)^m, & nT_i \leq t < (n+1)T_i \\ 0, & \text{otherwise} \end{cases} \tag{3.234c}$$

Application of the continuous-time Fourier transform to the basis functions $g'(m,n,t)$ yields

$$G'(m,n,\omega) = \frac{1}{T_i} \mathscr{F}\left\{g'(m,n,t)\right\}$$

$$= \begin{cases} 2\cos\left(\left[n + \frac{1}{2}\right]\omega\right)\left((-1)^{\frac{m}{2}} m! \Phi(m,\omega) + \mathrm{sinc}\left(\frac{\omega}{2}\right)\right), & m \text{ even} \\ 2\sin\left(\left[n + \frac{1}{2}\right]\omega\right)(-1)^{\frac{m+1}{2}} m! \Phi(m,\omega), & m \text{ odd} \end{cases} \tag{3.235}$$

where the function $\Phi(m,\omega)$ is defined as

$$\Phi(m,\omega) = \sum_{k=0}^{\lfloor (m-1)/2 \rfloor} \left(\frac{\omega}{2}\right)^{2k-m} \frac{(-1)^k}{(2k)!} \left(\frac{\sin\left(\frac{\omega}{2}\right)}{\frac{\omega}{2}} - \frac{\cos\left(\frac{\omega}{2}\right)}{2k+1}\right). \tag{3.236}$$

It is noted that this representation uses the normalized angular frequency ω (3.217) in contrast to the ordinary frequency variable f utilized in the original derivation. As a consequence of this normalization, the amplitude scaling due to the input sampling period T_i is also included in the basis functions (3.235).

In this way, the continuous frequency response of the modified Farrow structure can be stated as

$$H_c(\mathrm{j}\omega) = T_i \sum_{m=0}^{M} \sum_{n=0}^{\frac{N-1}{2}} c'_{mn} G'(m,n,\omega). \tag{3.237}$$

However, the representation of $\Phi(m,\omega)$ (3.236) is undefined for $\omega = 0$ and is numerically unstable if $|\omega|$ is small. For this reason, a formula based on Maclaurin series expansions of the sine and cosine terms is stated in [VS07]

$$\Phi(m,\omega) = \sum_{k=0}^{\lfloor \frac{m-1}{2} \rfloor} \sum_{l=\lfloor \frac{m-1}{2} \rfloor + 1}^{\infty} \left(\frac{\omega}{2}\right)^{2(k+l)-m} \frac{(-1)^{k+l} 2(k-l)}{(2k+1)!(2l+1)!}. \tag{3.238}$$

For evaluation, the infinite summation has to be replaced by an appropriate finite upper limit. Using this approximation for low frequencies decreases numerical errors. However, as the summation contains negative powers of ω, this approximation is still undefined for $\omega = 0$ and numerical cancellation is likely to occur for very small arguments due to the alternating signs of the addends.

Because the frequency response is linear with respect to the filter coefficients c'_{mn}, many efficient optimization techniques are directly applicable. In [Ves99, VS07], weighted least-squares

and minimax optimization design methods are proposed. Both optimization methods utilize a discrete grid of frequencies spanning both the passband and the stopband of the approximation region (3.220). Additional time domain conditions, specifically the interpolation condition, the continuity of the impulse response $h_c(t)$ and a continuous first derivative of $h_c(t)$, are introduced in [VS07]. For that purpose, the optimization problem is transformed into a different unconstrained least squares or minimax problem, respectively. The transformed problem has a reduced number of variables that determine the coefficients c'_{mn} uniquely.

An alternative frequency-domain description for the original form of the modified Farrow structure (3.117) has been proposed in [Ves00]. Starting from a discrete representation of $h_c(t)$ obtained by sampling the continuous impulse response with a sampling frequency K/T_i, $K \in \mathbb{N}$, approaching the limit $K \to \infty$ yields an expression for $H_c(j\omega)$

$$H_c(j\omega) = e^{-j\omega N/2} T_i \sum_{m=0}^{M} C_m(\omega) W_m(\omega). \tag{3.239a}$$

Here, $C_m(\omega)$ denotes the zero-phase discrete-time frequency response of the Farrow subfilters $C_m(z)$. $W_m(\omega)$ is a weighting function associated with each polynomial order m

$$C_m(\omega) = \begin{cases} 2\sum_{n=0}^{(N-1)/2} c'_{mn} \cos\left(\left[n+\frac{1}{2}\right]\omega\right), & m \text{ even} \\ 2\sum_{n=0}^{(N-1)/2} c'_{mn} \sin\left(\left[n+\frac{1}{2}\right]\omega\right), & m \text{ odd} \end{cases} \tag{3.239b}$$

$$W_m(\omega) = \begin{cases} \sum_{k=0}^{m} \binom{m}{k} \frac{\sin(\omega/2+k\pi/2)}{(\omega/2)^{k+1}}, & m \text{ even} \\ \sum_{k=0}^{m} \binom{m}{k} \frac{\cos(\omega/2+k\pi/2)}{-(\omega/2)^{k+1}}, & m \text{ odd} \end{cases} . \tag{3.239c}$$

This representation establishes a connection between the discrete-time frequency responses of the subfilters $C_m(z)$, represented by the corresponding zero-phase response $C_m(\omega)$, and the continuous-time frequency response $H_c(j\omega)$. That is, neglecting scaling and the implementation delay to ensure causality, $H_c(j\omega)$ is formed by combining the periodic frequency responses of the subfilters $C_m(z)$ weighted by a nonperiodic function $W_m(\omega)$.

This representation also clarifies the properties of continuous frequency responses obtained from symmetric impulse responses as characterized in Section 3.8.1. Apart from the implementation delay $e^{-j\omega N/2}$, $H_c(j\omega)$ is a real function. Moreover, it is observed that the weighting functions $W_m(\omega)$ are even or odd functions for m even or odd, respectively. Since the zero-phase frequency responses of the subfilters $C_m(\omega)$ have the same parity, it follows from (3.239a) that $H_c(j\omega)$ is an even, i.e. symmetric, function.

3.8.3 Design of ASRC Systems Incorporating Rational SRC

As depicted in Figure 3.42, ASRC systems incorporating integer-ratio SRC consist of two distinct components, the discrete-time prefilter $H_{dig}(e^{j\omega})$ and the continuous-time resampling function $H_{int}(j\omega)$. Thus, coefficient design has to be performed for these two components. In existing approaches, these components are typically considered and designed independently.
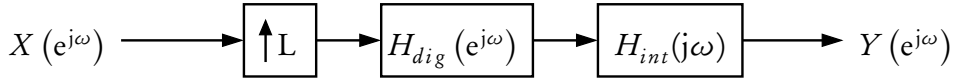
Figure 3.42: ASRC structure based on integer-ratio sample rate increase and a continuous resampling function, frequency-domain representation.

3.8.3.1 Design of the Discrete-Time Anti-Imaging Filter

If the components are designed independently, the discrete-time prefilter $H_{dig}(e^{j\omega})$ forms an anti-imaging filter of an integer-ratio sample rate converter with oversampling ratio L. Thus, the ideal frequency response follows from (3.60)

$$\widehat{H}_{dig}\left(e^{j\omega}\right) = \begin{cases} L, & |\omega| < \frac{\pi}{L} \\ 0, & \text{otherwise} \end{cases}. \tag{3.240}$$

Design specifications for practical anti-imaging filters as well as corresponding design methods are described in detail in standard texts on rational sample rate conversion, for instance [SR73, CR81, CR83].

As stated in Section 3.4.2, FIR filters are preferable for most SRC applications. For rational SRC, standard FIR design methods based on window or equiripple design methods as well as special designs such as halfband filters are widely used.

In [Eva00b], the design of discrete-time anti-imaging filters for use in ASRC systems is investigated. The design methods in this work include frequency-domain least-squares approximation, peak-constrained least squares, and the design of L-th band filters. Nonetheless, the objective functions are identical to the filter specifications for conventional, rational SRC.

3.8.3.2 Design of the Continuous-Time Resampling Filter

In most approaches to ASRC incorporating integer-ratio oversampling, classical interpolation or approximation methods from numerical analysis are used as continuous-time resampling functions (e.g. [LPW82, Ram84]). Evaluations of the performance of various numerical interpolation formulas, including Lagrange interpolation, Hermite or osculatory interpolation [Gre44, Sch46a], and B-spline basis functions in combination with oversampling have been performed in [ZB94, WBJ99]. These investigations suggest that the passband performance of Hermite and osculatory interpolation is superior to the other methods at the cost of a degraded image attenuation. In contrast, the stopband performance of B-spline functions exceeds that of Lagrange interpolation, but requires a compensation in the passband.

The design of continuous-time functions with respect to time- or frequency-domain conditions is considered in [Eva00a, Eva00b]. Therein, the continuous-time resampling functions are modeled either as single polynomials or as piecewise polynomial functions. The time-domain optimization techniques for these classes of resampling functions have already been characterized in Section 3.8.2.2, because they are independent of the presence of oversampling.

[Eva00b, Eva03] also considers the design of $H_{int}(j\omega)$ with respect to least-squares and minimax frequency-domain objectives. However, the design specification of $H_{int}(j\omega)$ does not

consider the oversampled nature of the input signal. Thus, the design is identical to piecewise polynomial resampling functions considered in Section 3.8.2.

In this way, the characteristics of the overall structure, for instance the frequency scaling due to oversampling or the role the discrete-time prefilter, are not utilized in the design of $H_{int}(\mathrm{j}\omega)$. While [Eva00b] suggests to adapt the characteristics of the continuous-time resampling function in order to correct passband deviations caused by the prefilter $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$, neither design methods nor examples are given. In any case, the practical utility of this approach appears rather limited, because the continuous-time resampling function is typically the most costly part in an ASRC system [BTU99, BTU01]. Consequently, compensation of the passband response can be implemented more efficiently in the discrete-time prefilter than in the continuous-time resampling function.

In summary, the design methods proposed so far consider $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$ and $H_{int}(\mathrm{j}\omega)$ as independent entities. While allowing the use of well-established design techniques for both components, this approach has two main shortcomings. First, the independent designs do not take the particular characteristics of the respective other component into account. Second, the partitioning into independent design problems prohibits purposeful optimization methods to determine the optimal parameterization for a given error norm and additional design specification.

3.9 Comparison and Relations between FD Filtering and ASRC

In this chapter, FD filtering and ASRC have been characterized as two related, but distinct approaches to delay discrete-time sequences by arbitrary time values. It is therefore worthwhile to compare these approaches both from a conceptual perspective and from a practical point of view concerned with design and application issues.

Handling of Delay Changes   FD filtering is a purely discrete-time filtering operation. While VFD filters enable changes of the delay value, only discrete changes are considered. As characterized in Section 3.3.3, these changes give rise to two potential effects: transients and signal discontinuities. Transients occur only for IIR FD filters or if the implementation structure uses intermediate results that are based on past delay values.

The discontinuities result from the discrete change of the filter characteristic and are therefore inherent to the FD approach. Because such artifacts are highly audible, remedies such as a sequence of small, gradual changes [Väl95a] or a crossfade between two signals delayed by the old and new value [Smi10a] have been proposed. However, these techniques generally require additional resources. Moreover, while minimizing the audible effects due to the discontinuities, these algorithms do not alter the discrete nature of parameter changes.

In contrast, ASRC systems generally utilize a different intersample position, corresponding to the fractional delay value of VFD system, for each processed output sample. Thus, the requirement to handle varying delay values is intrinsic to ASRC algorithms. In particular, the support of arbitrary, slowly time-varying conversion ratios has been characterized in Section 3.4.3 as a distinguishing feature of ASRC. Applied to delaying discrete-time sequences, this ability corresponds to the handling of delay trajectories that are continuous, smooth functions of time.

Analytical Relations   A formal relationship between FD filtering and ASRC can be established by relating the respective system responses. Such a connection, expressing the discrete-time frequency response $H(e^{j\omega}, \mu)$ of a VFD filter in terms of the continuous-time frequency response $H_c(j\Omega)$, has been first published in [VS96b].

As stated in Section 3.5.3, VFD as well as ASRC systems implemented as finite discrete convolutions are conveniently expressed by a single continuous-time function $h_c(t)$. In case of ASRC, $h_c(t)$ forms the continuous-time impulse response of the resampler. For VFD applications, the discrete sequence of filter coefficients $h(k, \mu)$ is obtained from $h_c(t)$ by shifting and sampling (3.98)

$$h(k, \mu) = h_c \left(\left[\mu - t_k \right] T_i \right) .$$

This sampling operation can be expressed by modulating $h_c(t + \mu T_i)$ by a continuous-time impulse train $s(t)$ (3.1) with period T_i.

$$h(k, \mu) = s(t) h_c(t + \mu T_i). \tag{3.241}$$

Applying a continuous Fourier transform yields

$$H(j\Omega, \mu) = \frac{1}{2\pi} S(j\Omega) * \mathscr{F} \{h_c(t)\} \tag{3.242a}$$

$$= \frac{1}{T_i} \sum_{k=-\infty}^{\infty} \delta \left(\Omega - k\Omega_i \right) * \left[e^{-j\Omega\mu} H_c(j\Omega) \right] \tag{3.242b}$$

$$= \frac{1}{T_i} \sum_{k=-\infty}^{\infty} e^{-j(\Omega - k\Omega_i)\mu} H_c \left(j \left[\Omega - k\Omega_i \right] \right) . \tag{3.242c}$$

Here, $S(j\Omega)$ denotes the Fourier transform of the impulse train $s(t)$ defined by (3.8). The discrete frequency response is obtained by normalizing the angular frequency to the input sampling period T_i

$$H(e^{j\omega}, \mu) = H(j\Omega, \mu)\big|_{\Omega = \frac{\omega}{T_i}} \tag{3.243a}$$

$$= \frac{1}{T_i} \sum_{k=-\infty}^{\infty} e^{-j(\omega - 2\pi k)\mu} H_c \left(\frac{\omega}{T_i} - \frac{2\pi}{T_i} k \right) . \tag{3.243b}$$

Thus, the discrete-time frequency response of the VFD filtering structure is formed as a superposition of periodic repetitions of $H_c(j\Omega)$, each subject to an individual phase shift depending on the intersample position.

Several useful conclusions can be drawn from this representation. First, it is obvious that an ideal ASRC filter implies an ideal FD element, because all signal images except for $k = 0$ vanish. For realizable filter structures, equation (3.243) clarifies how the principal errors of ASRC systems, that is, passband deviations and signal images, result in errors of the corresponding FD element. In particular, incompletely attenuated images in $H_c(j\omega)$ are aliased into the discrete-time frequency response $H(e^{j\omega}, \mu)$. Consequently, a good performance in the ASRC sense implies good quality with respect to FD criteria. However, the converse is not necessarily true.

Finally, the interpretation as an ASRC system enables a separation of qualitatively different errors of a VFD filter. The error corresponding to the central image of $H_c(j\Omega)$, that is, for

$k = 0$, represents a pure amplitude deviation independent of the intersample position μ. In contrast, the errors due to incompletely attenuated signal images, i.e. $k = \pm 1, \pm 2, \ldots$, fluctuate as a function of the intersample position and introduce phase errors. This separation is beneficial since constant, static inaccuracies are often less severe than varying errors. Moreover, a static amplitude deviation can often be compensated in the overall signal flow of a processing system. In this way, the interpretation as an ASRC system could potentially be used for the design of VFD structures.

Design Aspects   In the Sections 3.7 and 3.8, numerous design criteria and methods for VFD filters and ASRC systems have been described. Generally, design of VFD filters is a complex-valued two-dimensional approximation problem with a bounded approximation region spanned by the periodic frequency variable ω and a unit-sized interval of the fractional delay value or the intersample position. Utilizing symmetry relations of the filter halves the number of design parameters as well as the size of the approximation region.

In contrast, ASRC design problems translate to one-dimensional design problems with the frequency variable Ω (or its normalized representation ω) as independent variable. While the approximation range for Ω is infinite in theory, finite limits are sufficient in practice. Moreover, as symmetric system responses are assumed in all relevant ASRC designs in use, the approximation problem is real-valued. Like in case of VFD design, this choice halves the size of the approximation region and the number of free parameters.

As evident from these characterizations, ASRC design problems are typically solved more efficiently and are less susceptible to numerical inaccuracies than comparable VFD designs. First, the description as a real-valued optimization problem is more straightforward and more efficient to solve. Second and more important, the size of the optimization problem is typically much smaller for ASRC. In particular, if the approximation region is discretized, the two-dimensional grid for VFD designs results in a very large number of discrete points, requiring more computational resources for optimization and potentially increasing numerical conditioning problems.

3.10 Conclusions

In this chapter, the problem of delaying discrete-time sequences by arbitrary values has been reviewed and systematized. Two main approaches are common in DSP: Variable fractional delay filtering (VFD) and sample rate conversion, with arbitrary sample rate conversion (ASRC) as its most general form. Although both approaches base on the concepts of sampling and reconstructing bandlimited signals, they also show significant distinctions. While VFD is a discrete-time filtering process that can be controlled by an additional delay parameter, ASRC is best modeled as a continuous-time process.

Despite these conceptual differences, both approaches are typically implemented as discrete-time convolution processes. Consequently, they can be described within a uniform framework, easing comparison and enabling a common representation of algorithms. Actually, the algorithms and implementation structures used to implement both approaches are identical in most cases. In contrast, the design methods to parameterize these structures are profoundly different for ASRC and VFD. These distinctions comprise design objectives, the form of the approximation problem, and the design procedures in use.

Analytical relations show that a good performance in the ASRC sense implies good quality with respect to VFD error measures. However, the way delay changes are handled appears to be the most important distinction between both approaches. While, from a conceptual point of view, VFD enables only discrete changes of the delay parameter, the ability to reproduce continuously time-varying delays is intrinsic to arbitrary sample rate conversion.

Chapter 4

Signal Processing for Wave Field Synthesis

4.1 Introduction

In Chapter 2, delaying and scaling of discrete-time signals has been identified as a major signal processing operation in a WFS reproduction system. Chapter 3 introduced two models to describe arbitrary delay operations on discrete-time signals as well as a number of algorithms and implementation structures for this operation.

In order to develop an efficient signal processing system for WFS, several questions remain to be solved. In Section 4.2, it is argued why arbitrary sample rate conversion is more appropriate than variable fractional delay filtering to characterize the delay operations in a WFS system.

Complexity issues of WFS signal processing systems are discussed in Section 4.3. To mitigate the excessive complexity growth for real-world reproduction systems, a processing scheme that can be used with several efficient implementation structures for ASRC is proposed.

Section 4.4 investigates potential aliasing artifacts due to time-variant delays. It discusses conditions that require an explicit handling of this problem. For cases that require sophisticated anti-aliasing functionality, a solution is proposed that utilizes the efficient processing scheme proposed in Section 4.3 and causes only a moderately increased complexity.

4.2 An Appropriate Model for the Delay Operation

As characterized in the preceding chapter, two different models to describe time-variant delay operations on discrete-time sequences are predominantly used: variable fractional delay filtering (VFD) and arbitrary sample rate conversion (ASRC). It is therefore necessary to determine and justify which model is more appropriate for WFS signal processing.

For the reasons explained in Section 2.5, the necessity for a dedicated handling of time-variant delays arises from the reproduction of moving sound sources. The movement of the sound sources results in a time-variant distance between the source and each loudspeaker. Thus, the delay value calculated by the WFS operator is continuously time-varying.

As argued in Section 3.9, VFD filters do not consider the effects of continuous changes to the delay value. In contrast, the handling of continuously varying intersample positions is an intrinsic requirement of sample rate conversion systems, in particular ASRC.

On the other hand, virtual sound sources in WFS typically represent physical real-world objects which move with a bounded acceleration. That is, the source velocity has a limited rate of change. Consequently, the rate of change of the corresponding WFS delays is also bounded. In this way, the delay operation is appropriately modeled by a slowly time-varying sample rate conversion process. As described in Section 3.4.3, the support of such slowly time-varying conversion ratios is a distinguishing feature of ASRC algorithms.

From the physical point of view, the primary effect of a moving sound source is the Doppler effect (Section 2.4.1), resulting in a shift of the frequency contents at a stationary receiver position. In a discrete-time signal, this frequency shift is equivalent to a change of the sampling periods underlying the involved discrete-time sequences. In this way, ASRC forms a discrete-time approximation of the Doppler shift that models the perceived effect of this physical process directly.

On the other hand, the performance metrics used for VFD filters are also meaningful for WFS reproduction systems. For instance, magnitude and phase distortion of loudspeaker driving signals should be confined within reasonable bounds for stationary virtual sources. Therefore, it is interesting to consider how much the performance metrics for VFD deteriorate if the time-variant delay algorithms for WFS are designed according to ASRC criteria. As shown in Section 3.9, the error metrics for VFD and ASRC are related. In particular, the performance metrics for ASRC determine the response in the VFD sense completely, while the opposite is not necessarily true. As a consequence, good performance with respect to ASRC design criteria implies reasonable quality according to VFD error measures.

It should be noted that algorithms designed in the VFD or ASRC sense are not completely different, but exhibit only specific distinctions. It is therefore an interesting idea to combine criteria of both models into a single design specification. While such approaches are outside the scope of the present work, they form a promising starting for future research, in particular when combined with quality criteria for audio applications.

In summary, this section pointed out several reasons why the time-variant delays required for WFS should be modeled as an ASRC problem rather than an application of variable FD filtering. In addition, this choice does not imply a significant loss of quality with respect to other design criteria.

It should be reemphasized that modeling the time-variant delay operation as an ASRC process does not change the general structure of WFS signal processing. It affects only the way in which loudspeaker signals are calculated from the delay-line structures containing the signals of virtual sound sources. From an external perspective, WFS rendering is invariably implemented as a single-rate discrete-time signal processing system.

4.3 Reduction of the Computational Complexity

As motivated in Section 2.3, scaling and delaying of discrete-time signals is typically the most performance-critical operation in a realtime WFS rendering system. In particular, the number of simultaneous delay operations is proportional both to the number of virtual sources and the number of loudspeakers. Because this product is typically quite large, the associated computational complexity prohibits the use of conventional high-quality ASRC algorithms in most cases.

4.3.1 Utilizing the Structure of WFS Signal Processing

One way to overcome the excessive complexity growth caused by the application of high-quality resampling algorithms is to take advantage of the structure of a WFS signal processing system.

Although the number of simultaneous delay operations is large, these operations are not completely independent. For instance, a large number of loudspeaker signal components are
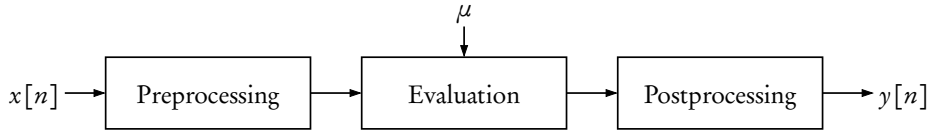
Figure 4.1: Proposed partitioning scheme for ASRC algorithms.

obtained from the signal of each virtual source. Likewise, each loudspeaker signal is formed as a linear combination of a multitude of scaled and delayed signals.

This structure can be utilized to reduce the overall computational complexity. To this end, operations that do not have to be performed for each combination of a virtual source and a loudspeaker are factored out of the ASRC algorithm. In this way, the WFS signal processing structure is partitioned into several distinct components. This partitioning scheme has been initially proposed in the context of WFS in [FBR08].

To put it another way, the reduction of the computational effort stems from considering the WFS signal processing structure as a multiple-input multiple-output (MIMO) system (see, e.g. [Vai92]) instead of a set of individual scale and delay operations.

4.3.2 Proposed Algorithm Partitioning

The proposed partitioning of the scale&delay operation for WFS is shown in Figure 4.1. The roles and requirements of the different stages are as follows:

Preprocessing The preprocessing stage transforms the input sequence $x[n]$ into a stream of intermediate data. The intention of this stage is to perform operations on the input data that allow the subsequent parts of the ASRC algorithm to operate more efficiently. The main requirement for this stage is that the performed operations are independent of the actual intersample position μ, and consequently of the resampling ratio R.

Evaluation This stage operates on the intermediate data produced by the preprocessing stage and generates outputs that depend on the intersample position μ. For this reason, this operation has to be performed for each combination of primary and secondary sources.

Postprocessing The postprocessing stage may apply additional transformations to the output values generated by the evaluation stage. As in case of the preprocessing stage, these operations must be independent of the output instants.

For a particular ASRC algorithm, it is not necessary that all stages actually perform nontrivial operations. For instance, many algorithms do not require an explicit postprocessing stage.

Integrating this partitioning scheme into the signal processing structure for WFS shown in Chapter 2, Figure 2.3, results in the adapted processing structure depicted in Figure 4.2.

The preprocessing stage is performed once for the signal of each virtual source, $x_m[k]$. The result of this operation is stored in a component termed *intermediate data* for each component m. While the source signals are written into a digital delay line in the original processing structure, the intermediate data values are not required to represent past input samples for the proposed partitioning scheme. In any case, the intermediate data components exhibit key features of delay lines, namely a multiple, arbitrary access to its contents based on the integer part of
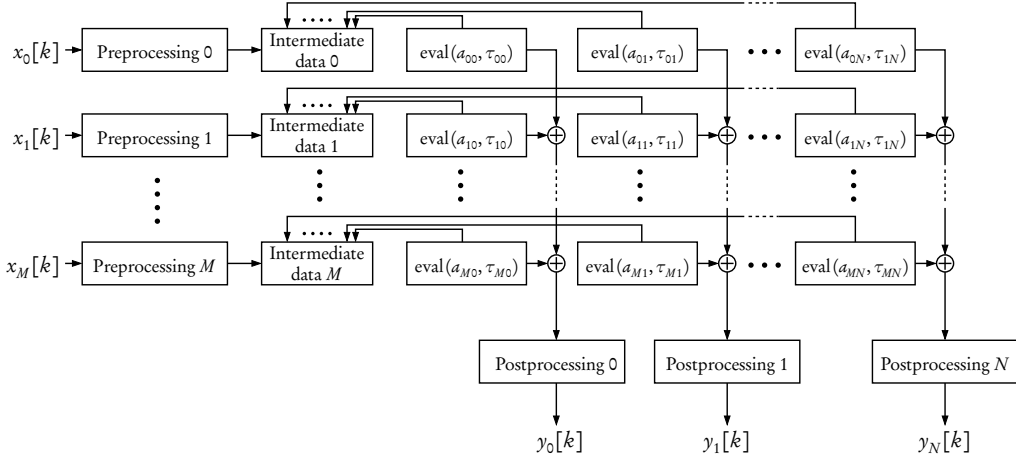
Figure 4.2: Proposed signal processing structure for WFS.

the desired delay value. The evaluation stage is performed for each combination of a virtual source and a loudspeaker signal. This is represented by the components $\mathrm{eval}(a_{mn}, \tau_{mn})$, which combine the delay operation with the amplitude scaling by a_{mn}. After accumulating the signal components according to the loudspeaker indices n, the postprocessing stage is performed for each loudspeaker signal.

It is noted that the WFS equalization filter $H(\omega)$ (2.5) present in Figure 2.3 is not explicitly shown in the proposed processing scheme. Instead, it is assumed that this filtering operation is incorporated either into the pre- or the postprocessing stage. The appropriate location for this operation can be determined based on several criteria. First, depending on the numbers of sources and loudspeakers, the computational effort might be lower if $H(\omega)$ is implemented either in the pre- or in the postprocessing stage. Second, the spectral equalization might be combined with the processing performed in one of these stages, resulting in a reduced overall complexity. Finally, if dedicated spectral correction filters are used that incorporate either source- or loudspeaker-dependent characteristics (e.g. [ARB04, SA10]), the variable equalization filters must be applied to either the source or the loudspeaker signals.

As motivated above, the main intention of this structure is to decrease the computational complexity of the overall algorithm. For this purpose, the performance of the evaluation stage is of utmost importance. By introducing pre- and postprocessing stages which might be relatively costly in terms of computational complexity, a significant reduction in the overall complexity is gained if these operations permit a more efficient implementation of the evaluation stage.

4.3.3 Suitability of ASRC Implementation Structures

In Section 3.6, numerous implementation structures for ASRC have been described. In the following, it is investigated how well these structures can be integrated into the proposed partitioning scheme. The main criterion is whether a relevant portion of the algorithm can be transposed to either the pre- or the postprocessing stage, resulting in a reduced complexity of the

evaluation stage.

For reasons that will be explained in detail in Section 4.4, implementation structures for sample rate reductions, such as the transposed Farrow structure, are not considered here.

Methods with Explicitly Calculated Filter Coefficients   Several resampling algorithms, for instance Lagrange interpolation (Section 3.6.3) or methods based on splines (Section 3.6.4), compute the discrete-time coefficients $h(n, \mu)$ of the resampling filter based on explicit formulas. As stated there, these computations depend on the intersample position μ and constitute a major part of the complexity of the algorithms. For this reason, no major processing task independent of the intersample position can be transposed into either the pre- or postprocessing stage. Thus, this class of implementation structures is not suited for the proposed partitioning scheme.

Methods based on Oversampling and Continuous-Time Resampling Functions   Methods incorporating integer-ratio oversampling as described in Section 3.6.5 consist of two cascaded components: A discrete-time sample rate increase and a continuous-time resampling function controlled by the intersample position μ. Therefore, this class of structures fits into the proposed partitioning scheme in a natural way. The integer sample rate increase constitutes the preprocessing stage, whereas the continuous-time resampling function represents the evaluation stage. Consequently, the storage components for intermediate data according to Figure 4.1 represent arbitrary-access delay lines containing oversampled representations of the source signals $x_m[k]$. Thus, the amount of intermediate data is increased by the oversampling ratio L compared to storage size of the delay line in the original processing scheme. This implementation structure does not require a dedicated postprocessing stage.

Methods Based on the Farrow Structure   As shown in Section 3.6.1, all variants of the Farrow structure, including the modified and the generalized form, are based on the evaluation of a set of $M + 1$ FIR filters, the so-called subfilters $C_m(z)$. The actual intersample position is incorporated only in the evaluation of a polynomial that uses the outputs of the subfilters as coefficients.

Therefore, the Farrow structure can be readily integrated into the proposed partitioning scheme. For this purpose, the preprocessing stage consists of the evaluation of the subfilters $C_m(z)$. For each input sample, the preprocessed data comprise the $M + 1$ output values of the subfilters. These values are stored in the intermediate data structure, a delay-line-like structure enabling arbitrary access based on the integer part of the required sample delay. The evaluation stage consists of the evaluation of the polynomial for the current value of μ.

Consequently, the storage required by the proposed partitioning scheme is $M + 1$ times the amount of memory required by a conventional WFS implementation structure. As in case of structures based on oversampling, this implementation does not require a nontrivial postprocessing stage.

4.3.4 Performance Comparison

To demonstrate the potential performance gains of the proposed algorithm partitioning, different implementation structures are compared in a scenario resembling a realistic WFS reproduction
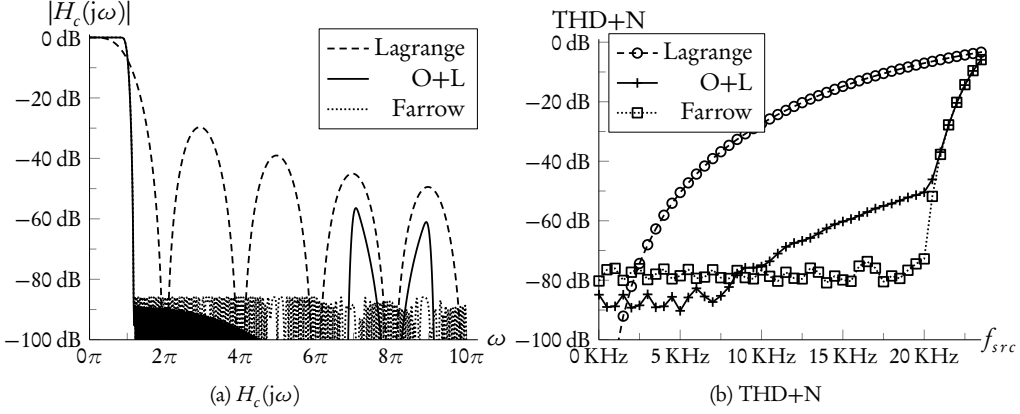
Figure 4.3: Quality of ASRC algorithms used in the performance comparison.

system. Here, a system consisting of $N = 256$ speakers operating at a sampling frequency of 48 KHz is considered. This setup corresponds roughly to a small to medium-sized movie theater.

Three different algorithms are compared. Cubic Lagrange interpolation, that is, of order $N = 3$, represents a resampling algorithm based on explicitly calculated filter coefficients. As stated above, this class of structures cannot gain a performance improvement from the proposed partitioning scheme. Cubic Lagrange interpolation is widely used in audio applications (e.g. [LVKL96, WBJ99]). Moreover, it often represents the most basic algorithm that provides acceptable audio quality because of the relatively severe errors of linear interpolation.

The second candidate comprises integer-ratio oversampling with ratio $L = 4$ and a cubic Lagrange interpolator of order $N_{dig} = 3$ as continuous-time resampling function. This algorithm is denoted as Oversampling+Lagrange (O+L). The prototype filter of the oversampling component is of order $N_{dig} = 127$. It is designed with respect to a L_∞ criterion with cutoff frequency $f_c = 20$ KHz.

A resampling algorithm based on the modified Farrow structure with orders $M = 5$, $N = 31$, and the same cutoff frequency designed according to the L_∞ norm forms the third candidate.

To compare the resampling quality of these algorithms, the continuous frequency responses $H_c(j\omega)$ are shown in Figure 4.3a. It is observed that the Lagrange interpolator exhibits a noticeable roll-off in the passband as well as a limited attenuation of passband images with a worst-case stopband error of -29.7 dB. In contrast, the Oversampling+Lagrange algorithm has very good image attenuation, but shows a pronounced signal image around $\omega = 8\pi = 2\pi L$, resulting in a maximum stopband error of about -56.4 dB. In contrast, the stopband error of the Farrow structure shows equiripple behavior over wide parts of the stopband region with a maximum error of -85.9 dB.

As a second performance measure, the THD+N (total harmonic noise+distortion) is utilized. The THD+N measure represents the ratio between the power of the signal error and the power of the signal. This measure is closely related to the signal-to-noise ratio (SNR), which is defined as the ratio between the desired signal and noise. The use of the THD+N measure in resampling

Algorithm	Preprocessing	Evaluation	Total operations	Memory size	Memory bandwidth
	operations per input sample	operations output sample	operations per output sample	data values per input sample	memory accesses per output sample
Lagrange	—	19	19	1	4
O+L	252	19	≈ 20	4	4
Farrow	282	10	≈ 11.1	6	6

Table 4.1: Computational effort for the partitioning scheme for WFS. Number of instructions/data values for computing one output sample.

applications is motivated in [Str00]. For a finite-length time-domain signal, it is calculated as

$$\mathrm{THD+N} = \frac{\mathscr{P}_e}{\mathscr{P}_x}, \tag{4.1}$$

where $\mathscr{P}_x$ denotes the average signal power of a discrete-time signal $x[n]$ (e.g. [Smi07])

$$\mathscr{P}_x = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2 . \tag{4.2}$$

Likewise, $\mathscr{P}_e$ denotes the average power of the error signal $e[n]$, that is, the difference to the desired, undistorted signal $\hat{x}[n]$

$$e[n] = x[n] - \hat{x}[n]. \tag{4.3}$$

In Figure 4.3b, the THD+N measures of the tested algorithms are shown as a function of the signal frequency of a virtual sound source. This source moves away from a stationary secondary source with a velocity $v = \frac{1}{5\pi}c \approx 21.6\,\mathrm{m/s}$. This irrational value results in a relatively dense spectrum of the noise. However, additional tests have shown that the noise power is relatively insensitive to the source velocity, that is, the conversion ratio.

As expected, the performance of the Lagrange interpolator is superior for low frequencies, but deteriorates relatively quickly as the input frequency increases. The Oversampling+Lagrange algorithm has an approximately constant error up to about 10 KHz, followed by a moderate increase up to the chosen cutoff frequency $f_c = 20\,\mathrm{KHz}$. In contrast, the THD+N of the Farrow structure is nearly constant in the whole passband interval of the design specification.

The computational effort of the different algorithms within the proposed partitioning scheme is shown in Table 4.1. The performance figures represent the effort to compute a single output sample. It is noted that the effort is independent of the conversion ratio, because the WFS signal processing structure forms a synchronous single-rate system from an external point of view.

As stated above, no explicit preprocessing stage is utilized in case of Lagrange interpolation. Therefore, the total effort per output sample is identical to the cost of the evaluation stage and amounts to 19 operations.

In contrast, the Oversampling+Lagrange algorithm and the Farrow structure perform an explicit preprocessing stage, consuming 252 and 258 instructions per input sample, respectively. The evaluation stage for the Oversampling+Lagrange structure is identical to the Lagrange interpolator. The evaluation part of the Farrow structure requires only $2M = 10$ instructions.

For the total complexity, the effort for the preprocessing stage is distributed over the $N = 256$ output samples. In this way, the complexity of the preprocessing is almost negligible, and the instruction counts amount to about 20 and 11.1 instructions, respectively. In case of the Oversampling+Lagrange algorithm, the effort is only slightly larger than the conventional Lagrange interpolator, while it is significantly lower for the Farrow structure. In this way, the proposed partitioning scheme enables a significantly higher resampling quality basically at the cost of the most basic conventional algorithms.

In the two rightmost columns of Table 4.1, the memory requirements of the different algorithms are shown. While the memory bandwidth per output sample remains constant or increases only moderately, the storage capacity of the algorithms with an explicit preprocessing stage is significantly larger than for the conventional Lagrange interpolator. Consequently, the proposed partitioning scheme might be regarded as a space-time trade-off. By using additional memory to store partial results, it enables reductions of the computational complexity by enabling an efficient reuse of intermediate data. Therefore, the utility of a particular algorithm depends on the characteristics of a target architecture. Depending on whether the available storage capacity or the memory bandwidth are scarce, different algorithms might prove superior.

4.4 Baseband Aliasing in WFS Signal Processing

4.4.1 Problem

As described in Section 3.4, algorithms for increasing and decreasing the sample rate are conceptually different. This distinction follows from the continuous frequency response of the ideal resampling filter (3.53)

$$\widehat{H}_c(\mathrm{j}\Omega) = \begin{cases} T_i, & |\Omega| < \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \\ 0, & |\Omega| \geq \min\left(\frac{\Omega_i}{2}, \frac{\Omega_o}{2}\right) \end{cases}. \tag{4.4}$$

For a sample rate increase ($\Omega_o > \Omega_i$), the cutoff frequency of $\widehat{H}_c(\mathrm{j}\Omega)$ is independent of the output frequency, while it depends on this rate for sample rate reductions ($\Omega_o < \Omega_i$). In the latter case, a lowpass filter with variable cutoff frequency is required to avoid aliasing, i.e. spectral foldover, of the baseband of the input signal into the baseband of the resampled signal. As motivated in Section 3.4, this type of aliasing is referred to as *baseband aliasing* to distinguish it from aliasing effects due to incompletely attenuated images.

The distinction between increasing and decreasing conversion ratios poses a conceptual problem for the application of ASRC algorithms in WFS. Because the need for sample rate conversion arises from moving sound sources, both sample rate increases and decreases must be supported. A source moving away from a secondary source corresponds to sample rate increase, i.e. a conversion ratio $R > 1$, which manifests itself in a decrease of the perceived frequency. Conversely, a source movement towards a secondary source implies a conversion ratio $R < 1$.

In practical WFS rendering systems, arbitrary increasing and decreasing conversion ratios may be required simultaneously. In Figure 4.4, a virtual source moving parallel to a linear array of secondary sources is depicted. For some secondary sources, such as $\mathbf{x}_{s0}$, the distance to the virtual sources is increasing, implying a conversion ratio $R > 1$. At the same time, the distance to other secondary sources, e.g. $\mathbf{x}_{s6}$ or $\mathbf{x}_{s10}$, is decreasing, corresponding to a ratio $R < 1$.
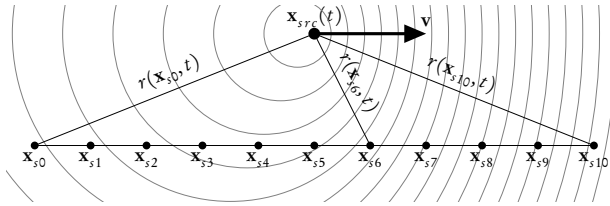
Figure 4.4: Different conversion ratios caused by the relative velocities with respect to the secondary sources.

4.4.2 Applicability of Existing Algorithms

The distinction between the sample rate increase case and the sample rate decreases case represented by the ideal frequency response (4.4) puts stringent requirements on algorithms used in WFS signal processing.

First, the vast majority of ASRC algorithms is limited to either sample rate increases or decreases. As an example, the transposed Farrow structure (Section 3.6.7) enables sample rate decreases by variable ratios, but is not suited for increasing the sample rate. Algorithms supporting increasing and decreasing sample rates, for instance [LK81a, LPW82, SG84], generally exhibit different modes for up- and downsampling. For applications with a large number of simultaneous ASRC operations such as WFS, such a conditional execution implies a more complex control logic.

This distinction also becomes apparent in implementations of variable-length delay lines proposed, for instance, in [Roc00, SSAB02]. If the conversion ratio is less than unity, then an interpolation operation has to be performed while writing to the delay line. The concept of such an *interpolating write* is related to the idea of deinterpolation (e.g. [Väl95a]), which enables the insertion of input samples at arbitrary instants into delay lines with a fixed sampling grid. However, deinterpolation has been proposed in a framework for digital waveguide modeling, that is, essentially single-rate systems. It is therefore not directly related to aliasing effects in SRC applications. Notwithstanding this difference, the interpolating write operation clarifies a major difference between algorithms for sample rate increases and decreases by flexible ratios [HF00, Hen02] (see also Section 3.6.7): While a sample rate increase corresponds to sampling a continuous impulse response $h_c(t)$ with the constant input sampling period T_i, decreasing the sampling rate corresponds to sampling $h_c(t)$ with the output period T_o.

Second, the algorithm partitioning scheme proposed in Section 4.3 places additional restrictions on the ASRC algorithms. The most stringent requirement is that the preprocessing stage must not depend on the intersample position or, equivalently, the conversion ratio. This condition inhibits the use of the transposed Farrow structure, because this algorithm does not possess a preprocessing stage independent of the conversion ratio. At the same time, algorithms based on oversampling and Lagrange interpolation or the Farrow structure are not applicable because they do not prevent baseband aliasing in case of sample rate decreases.

Finally, the concept of variable delay lines with interpolating writes illustrates a further implication of algorithms for sample rate decreases. If the ASRC algorithm operates on a delay-line like structure as in case of WFS, then the contents of the delay line depends on the conversion ratio if $R < 1$. Consequently, the contained signal cannot be reused by multiple read

operations with different conversion ratios. Thus, this class of algorithms cannot be used within the conventional processing structure for WFS shown in Figure 2.3, which uses a single delay line for each virtual source.

In summary, ASRC algorithms supporting decreasing conversion ratios cannot be integrated into WFS signal processing algorithms in a reasonable way. The lack of a preprocessing stage and of a data representation for preprocessed input signals, both independent of the conversion ratio, forms the main impediment.

4.4.3 Relevance for Practical Reproduction Systems

As shown above, the necessity to synthesize arbitrary sample rate decreases poses a conceptual difficulty for the use of ASRC algorithms within WFS. However, it is worthwhile to assess the impact of baseband aliasing artifacts in practical WFS rendering systems. In this section, conditions are determined under which WFS signal processing algorithms designed for nondecreasing conversion ratios may cause audible aliasing errors.

The main influencing factors for baseband aliasing errors in WFS are the maximum allowed source velocity, corresponding to a worst-case resampling ratio R_{min}, and the cutoff frequency ω_c of the input signal. The role of the cutoff frequency in SRC systems is described for instance in [CR83]. Its application to ASRC design is shown in Section 3.8.1. However, its role in WFS, especially for the assessment of baseband aliasing artifacts, requires a more sophisticated examination.

The perceived Doppler shift for a stationary listener is a special case of (2.11)

$$\omega_{rec} = \frac{c}{c - v} \omega_{src}. \tag{4.5}$$

Here, the source and receiver frequencies are given as normalized angular frequencies with respect to a common sampling frequency.

To model the Doppler shift as a sample rate conversion, ω_{src} and ω_{rec} are considered as representations of a single frequency Ω that are normalized to the sampling frequencies Ω_i and Ω_o, respectively

$$\Omega = \frac{\Omega_i}{2\pi} \omega_{src} \tag{4.6a}$$

$$= \frac{\Omega_o}{2\pi} \omega_{rec} = \frac{\Omega_o}{2\pi} \frac{c}{c - v} \omega_{src}. \tag{4.6b}$$

Thus, the conversion ratio is given by

$$R = \frac{\Omega_o}{\Omega_i} = \frac{c - v}{c}. \tag{4.7a}$$

Conversely, the Doppler shift can be expressed in terms of the conversion ratio as

$$\omega_{rec} = \frac{1}{R} \omega_{src}. \tag{4.7b}$$

It should be noted that (4.5) represents the true Doppler shift of a moving source. However, the analysis performed here is also applicable to WFS operators that neglect the notion of retarded time as described in Section 2.4. In this case, the conversion ratio R must be adapted appropriately.

The minimum conversion ratio R_{min} that may occur in a WFS reproduction system is determined by the maximum source velocity v_{max}

$$R_{min} = \frac{c - v_{max}}{c} .$$
(4.8)

This limit represents the worst-case scenario for baseband aliasing and corresponds to a virtual sound source $\mathbf{x}_\Psi$ moving towards a secondary source $\mathbf{x}_s$ with velocity v_{max}.

In general, baseband aliasing occurs if components of the spectrum $X\left(e^{j\omega}\right)$ of the source signal are translated into normalized frequencies exceeding π, which are aliased into the baseband of the output signal during resampling. As described in [CR83], two different conditions for baseband aliasing are widely used.

In the first case, aliasing is not allowed into the whole frequency interval $\omega \in [0, \pi)$. If the cutoff frequency ω_c denotes the highest frequency component in $X\left(e^{j\omega}\right)$, then this condition can be stated as

$$\frac{1}{R_{min}}\omega_c < \pi .$$
(4.9a)

Using (4.8), the maximum admissible source velocity that does not cause baseband aliasing is related to the cutoff frequency ω_c by

$$v_{max} < c \left(1 - \frac{\omega_c}{\pi}\right) .$$
(4.9b)

An alternative definition of baseband aliasing prohibits aliasing only into the frequency band of interest $[0, \omega_c]$, but permits frequency components to be folded into the transition band (ω_c, π) of the output signal $Y\left(e^{j\omega}\right)$. In this case, the condition to prevent aliasing reads

$$\frac{1}{R_{min}}\omega_c < 2\pi - \omega_c .$$
(4.10a)

Similar to (4.9b), the maximum admissible source velocity can be stated as

$$v_{max} < c\,\frac{2(\pi - \omega)}{2\pi - \omega} .$$
(4.10b)

For notational convenience, it is assumed that the highest frequency of interest of the output signal is identical to that of the input signal. This choice appears sensible for audio applications, where ω_c is typically related to the human hearing range. However, (4.10) can be extended to arbitrary combinations of cutoff frequencies without conceptual difficulty.

Naturally, the output signal is no longer bandlimited to ω_c if the conversion ratio falls below unity. The consequences of this behavior depend on the application. In some cases, the frequency

ω_c/π			0.5	0.6	0.7	0.8	0.9	1
f_c for $f_i = 48\,\mathrm{KHz}$	[KHz]		12	14.4	16.8	19.2	21.6	24
f_c for $f_i = 44.1\,\mathrm{KHz}$	[KHz]		11.03	13.23	15.44	17.64	19.85	22.05
No aliasing	[m/s]		170	136	102	68	34	0
Transition band aliasing	[m/s]		226.67	194.29	156.92	113.33	61.82	0

Table 4.2: Maximum admissible source velocity v_{max} in m/s to prevent baseband aliasing as a function of the cutoff frequency ω_c. Sound velocity: $c = 340$ m/s.

contents in the transition band $(\omega_c, \pi]$ can be ignored. In other applications, these aliasing components must be removed, for instance if subsequent signal processing components rely on explicit transition regions. In this case, the algorithm partitioning proposed in Section 4.3.2 can be utilized in a beneficial way. If the lowpass filter is implemented in the postprocessing stage of the processing scheme according to Figure 4.1, the output signals can be bandlimited to a user-defined cutoff frequency with relatively low impact on the overall computational effort.

In Table 4.2, the admissible source velocities are tabulated as a function of the cutoff frequency. It is apparent that for bandwidths typically used in audio applications (for example, $f_c \in [16\,\mathrm{KHz}, 20\,\mathrm{KHz}]$), the allowed velocities are relatively high compared to the source speeds typically used in WFS reproductions. In these cases, ASRC algorithms for sample rate increases can be applied without introducing perceptible aliasing errors. Consequently, baseband aliasing does not pose a practical problem in most applications of WFS.

4.4.4 Processing Structure to Prevent Baseband Aliasing

As stated in the preceding section, baseband aliasing does not actually result in audible artifacts for source velocities and signal bandwidths occurring in most applications. However, it is possible that in special circumstances, in particular if very high velocities are required in combination with wideband signals, baseband aliasing may become an issue. Therefore, a processing structure is proposed that prevents baseband aliasing for a moderate increase of the computational effort.

As shown above, the width of the transition region $(\omega_c, \pi]$, determined by the cutoff frequency ω_c, is the key influencing variable for the maximum admissible source velocity v_{max}. Therefore, a natural approach to increase v_{max} is to enlarge this transition region. This can be accomplished by increasing the sampling rate of the input signal, performing the ASRC computation at the higher sampling rate and decreasing the sample rate in the output signal again.

This processing structure can be integrated into the partitioning scheme proposed in Section 4.3. In this way, all ASRC algorithms that fit into this scheme are applicable. As illustrated in Figure 4.5, a sample rate increase by the fixed ratio R_{fix} is put in front of the ASRC algorithm, while the output of this structure is converted back to the original sampling frequency using a fixed sample rate decrease with ratio $1/R_{fix}$. In this way, the processing rate of all stages of the internal ASRC algorithm is increased by the factor R_{fix}. The systems for the fixed sample rate increases and decreases can be combined with the pre- and postprocessing stage of the internal ASRC algorithms. Thus, these operations are performed only once for each input and output signal, respectively. In this way, the computational effort for the fixed-ratio SRC is relatively
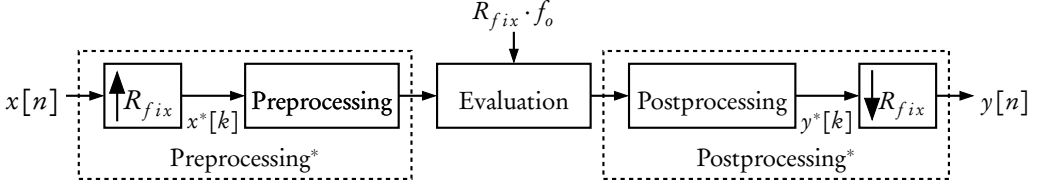
Figure 4.5: Proposed processing scheme to prevent baseband aliasing.

small compared to the overall complexity. Instead, the increase of the computational effort is mainly caused by the increased rate of computation for the evaluation stage, which is scaled by the ratio R_{fix}.

As stated above, the improvement of this processing structure results from widening the transition region or, equivalently, from the reduction of the cutoff frequency ω_c. For an input signal $x[n]$ with normalized cutoff frequency ω_c, the normalized cutoff frequency of the signal $x^*[k]$ after sample rate conversion by ratio R_{fix} is reduced to

$$\omega_c^* = \frac{1}{R_{fix}} \omega_c \,. \tag{4.11}$$

Applying this adapted cutoff frequency to (4.9b) and (4.10b) yields expressions for the maximum allowed source velocity in the modified processing structure

$$v_{max} = \begin{cases} c\left(1 - \frac{\omega_c}{R_{fix}\pi}\right), & \text{no aliasing allowed} \\ c\dfrac{1 - \frac{\omega_c}{R_{fix}\pi}}{1 - \frac{\omega_c}{2R_{fix}\pi}}, & \text{transition band aliasing allowed} \end{cases} . \tag{4.12}$$

In Figure 4.6, the maximum admissible source velocities are displayed as a function of ω_c and the fixed SRC ratio R_{fix}. The case $R_{fix} = 1$ corresponds to the conventional WFS signal processing structure without explicit prevention of baseband aliasing. It is seen that for a given ω_c, the admissible source velocity increases considerably even for a relatively small sample rate increase R_{fix}.

As a numerical example, consider the design specifications $\omega_c = 0.9\pi$ and $v_{max} = 150\,\mathrm{m/s}$ for a WFS reproduction system. According to (4.9b) and (4.10b), the conventional signal processing structure yields maximum source velocities $v_{max} = 34\,\mathrm{m/s}$ if aliasing is prohibited or $v_{max} \approx 61.8\,\mathrm{m/s}$ if aliasing in the transition region is acceptable. To fulfill the design specification $v_{max} = 150\,\mathrm{m/s}$, (4.9b) and (4.10b) are solved for the fixed resampling ratio, yielding $R_{fix} \approx 1.61$ and $R_{fix} \approx 1.26$ for the different aliasing conditions, respectively. As the overall computational load of the WFS signal processing structure is roughly proportional to the resampling ratio R_{fix}, aliasing prevention for these stringent design specifications can be implemented with an increase in complexity of about 61 % or 26 %, respectively.

It is worth emphasizing that the proposed processing structure for the prevention of baseband aliasing is distinct from ASRC algorithms utilizing fixed-ratio oversampling as described in
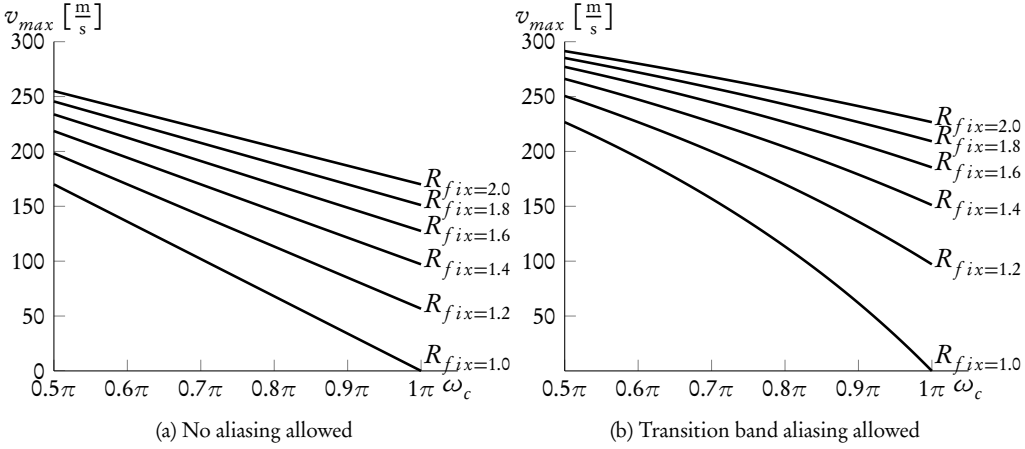
Figure 4.6: Maximum admissible source velocity in the processing scheme to prevent baseband aliasing.

Section 4.3.3. In conventional ASRC systems, the continuous-time resampling filter $H_{int}(j\omega)$ operates at the output sampling rate, and a separate sample rate decrease is not required at the output. Instead, in the structure proposed here, the continuous-time resampling function, corresponding to the evaluation stage of the partitioning, deliberately operates at a higher rate to increase the width of the transition band, consequently preventing aliasing into the baseband of the original signal. Thus, a separate sample rate decrease is necessary at the output of the system. However, this operation replaces the lowpass filtering with an arbitrary, variable cutoff frequency required in an ASRC algorithm for sample rate decrease by a fixed, rational SRC component that is executed only once for each loudspeaker signal.

4.5 Conclusions

In this chapter, efficient discrete-time structures for the audio signal processing within WFS reproduction systems have been considered. It turned out that scaling and delaying of audio signals is the most demanding signal processing operation, both in terms of the computational complexity and in its effect on the perceived quality of moving sound sources. It has been argued why this delay operation is best modeled as a time-variant arbitrary sample rate conversion operation.

However, due to the large number of simultaneous scale and delay operations in a typical WFS reproduction system, direct application of high-quality ASRC algorithms causes an excessive increase of the computational effort that is not feasible for most rendering systems. However, it is possible to exploit the structure of a WFS signal processing system to reduce the complexity of the delay operations significantly. For that purpose, a partitioning scheme for resampling algorithms is proposed that enables an efficient reuse of intermediate data.

It is shown that two important classes of efficient resampling algorithms, namely methods

based on the Farrow structure and algorithms incorporating integer-ratio oversampling, fit well into to this partitioning scheme. In this way, the proposed processing scheme enables the use of high-quality resampling algorithms basically at the cost of most basic methods for digital delay.

Because the time-variant delays in a WFS signal processing systems arise from arbitrarily moving sound sources, resampling algorithms must handle arbitrary increases and decreases of the sampling frequency. As shown in Chapter 2, these cases require conceptually different algorithms.

It turns out that conventional algorithms for decreasing the sample rate cannot be incorporated into efficient signal processing schemes for WFS reasonably. While decreasing sampling ratios potentially result in baseband aliasing artifacts, it is shown that this effect does not cause perceptible errors for the majority of use cases of WFS. For the remaining cases, a processing structure compatible with the proposed algorithm partitioning is developed that enables nearly arbitrary maximum source velocities and signal bandwidths. This modification increases the overall computational effort only moderately.

Notwithstanding the complexity reduction due to the proposed processing schemes, efficient ASRC algorithms remain a crucial component for WFS rendering systems. Additionally, the efficient processing structures proposed in this chapter pose new requirements and design objectives for ASRC algorithms. First, the suitability for the proposed algorithm partitioning forms a key requirement. Second, since the proposed processing schemes basically represent space-time trade-offs, additional metrics for the computational effort, such as storage size or memory bandwidth requirements, become increasingly important.

Chapter 5

Improvements to ASRC algorithms

5.1 Introduction

The importance and the widespread use of algorithms for arbitrary sampling rate conversion have been motivated in Section 3.4. Its application to WFS signal processing has been described in Chapter 4. For these reasons, algorithms that meet prescribed performance characteristics with minimal computational complexity are of great importance.

Existing research on ASRC algorithms most often focuses on implementation structures. However, the vast majority of algorithms requires a set of design parameters, such as filter orders, polynomial orders or oversampling ratios, to be specified. These decisions influence the performance and the computational complexity of the structures. Moreover, most ASRC structures exhibit a set of coefficients. Sophisticated design methods, resembling methods for discrete-time filter design, are required to determine these coefficients.

In order to compare different algorithms for ASRC, implementation structures cannot be assessed apart from design methods that yield suitable, preferably optimal coefficients for these structures with respect to a given design specification.

Often, ASRC algorithms consist of several building blocks, as in case of structures consisting of oversampling and continuous-time resampling filters. These components are typically considered independently, thus complicating design and analysis of the overall structure.

Generally, the advancements proposed in this chapter fall into two interrelated categories. First, analytical descriptions for the continuous frequency response of several classes of ASRC algorithms are derived. These expressions are directly applicable to design methods, but they also provide a better understanding of these algorithms.

The second focus is on design for these structures, targeting at methods that yield optimal quality with respect to selectable error norms. In addition, emphasis is put on flexible methods that enable the incorporation of additional requirements and constraints into the design process. Such flexible specifications facilitate the synthesis of ASRC designs that are specifically matched to the requirements of an application. For ASRC structures that consist of multiple components, it is investigated under which conditions such purposeful, optimal designs are achievable.

The improvements proposed in this chapter primarily target algorithms for sample rate increases. At the one hand, this restriction is justified by the application, as the efficient structures for WFS signal processing require this class of algorithms. On the other hand, most coefficient design methods are also applicable to sample rate decreases if the corresponding transposed form of the implementation structure is used.

5.2 A Unified Frequency Response for Symmetric Polynomial-Based Resampling Functions

Symmetric piecewise polynomial functions constitute a very general class of resampling filters. Therefore, analytic expressions for the frequency response of this class of functions are useful for a large number of ASRC algorithms. Moreover, as symmetric piecewise polynomial functions are efficiently implemented by the modified Farrow structure, exact representations of this frequency response are directly applicable to design methods for this implementation structure.

In Section 3.8.2, closed-form expressions for the continuous frequency response of the modified Farrow structure have been presented which were originally proposed in [Ves99, Ves00, VS07]. However, these representations exhibit several limitations. First, the derivations proposed so far are limited to odd subfilter orders. Moreover, they are undefined or prone to numerical instabilities for specific frequency arguments, for instance in the vicinity of zero. Finally, the complex structure of the existing representations complicates an intuitive interpretation of the properties of this class of resampling filters.

For these reasons, an alternative derivation is presented here. Emphasis is put on a uniform representation that holds for both even and odd subfilter orders. This derivation results in a concise form that is well-defined on the whole frequency axis and enables efficient and stable evaluation.

5.2.1 Derivation

As shown in Section 3.6.1, the continuous frequency response of piecewise polynomial basis functions is conveniently expressed as a linear combination of basis functions $f(m, \mu)$ (3.112)

$$h_c(t) = \sum_{m=0}^{M} \sum_{n=0}^{N} c_{mn} f\left(m, \tfrac{t}{T_i} + t_n\right) \text{ with} \tag{5.1}$$

$$f(m, \mu) = \begin{cases} \mu^m, & -\mu_{min} \leq \mu < \mu_{max} \\ 0, & \text{otherwise} \end{cases}.$$

The translation by a basepoint value t_n is conveniently expressed as

$$f\left(m, \mu + t_n\right) = \begin{cases} (\mu + t_n)^m, & -\mu_{min} - t_n \leq \mu < \mu_{max} - t_n \\ 0, & \text{otherwise} \end{cases}. \tag{5.2}$$

Due to the symmetry relations between the elements of the coefficient matrix c_{mn} (3.120a), the continuous impulse response $h_c(t)$ (5.1) can be expressed using the $(M+1) \times (N'+1)$ elements of the matrix B (3.129) and a set of modified basis functions $g(m, n, \mu)$

$$h_c(t) = \sum_{m=0}^{M} \sum_{n=0}^{N'} b_{mn} g\left(m, n, \tfrac{t}{T_i}\right). \tag{5.3}$$

The modified index limit N' has been defined in (3.123). The modified basis functions $g(m, n, \mu)$ are similar to those introduced in [Ves99, VS07], which are denoted as $g'(m, n, t)$ (3.234c) in

Section 3.8.2. However, the form used here is suitable for arbitrary orders irrespective of the parity of N, and does not require a transformation of the intersample position μ.

To determine these basis functions, three distinct cases must be considered. The first case applies if the order N is odd, or if N is even and $n > 0$. Then $g(m, n, \mu)$ consists of two polynomial pieces $f(m, \mu)$ that are symmetric with respect to zero.

$$g(m,n,\mu) = f\left(m, \mu + t_{N'-n}\right) + (-1)^m f\left(m, \mu + t_{N-N'+n}\right) \tag{5.4a}$$

$$= \begin{cases} (\mu + t_{N'-n})^m, & \mu_{min} - t_{N'-n} \le \mu < \mu_{max} - t_{N'-n} \\ (-1)^m (\mu + t_{N-N'+n})^m, & \mu_{min} - t_{N-N'+n} \le \mu < \mu_{max} - t_{N-N'+n} \\ 0, & \text{otherwise} \end{cases} \tag{5.4b}$$

Using the definition of the symmetric basepoint set (3.89) and the associated intersample position range (3.89d), (5.4b) can be specialized to even and odd subfilter orders N. For odd N, that is $N' = \frac{N-1}{2}$, $g(m, n, \mu)$ can be simplified to

$$g(m,n,\mu) = \begin{cases} (\mu + n + \frac{1}{2})^m, & -n - 1 \le \mu < -n \\ (-1)^m (\mu - n - \frac{1}{2})^m, & n \le \mu < n + 1 \\ 0, & \text{otherwise} \end{cases} \quad \text{for } N \text{ odd}. \tag{5.5a}$$

This basis function is depicted for even and odd polynomial orders m in the Figures 5.1a and 5.1b. Likewise, for even N, $N' = \frac{N}{2}$, it follows

$$g(m,n,\mu) = \begin{cases} (\mu + n)^m, & -n - \frac{1}{2} \le \mu < -n + \frac{1}{2} \\ (-1)^m (\mu - n)^m, & n - \frac{1}{2} \le \mu < n + \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \quad \text{for } N \text{ even and } n > 0. \tag{5.5b}$$

This case is illustrated in Figures 5.1c and 5.1d for even and odd m, respectively.

The second case holds if the filter order N is even, $n = 0$, and m is an even number. Then, $g(m, n, \mu)$ consists of a single polynomial piece that is symmetric with respect to $\mu = 0$.

$$g(m, 0, \mu) = f(m, \mu + t_{N'})$$

$$= \begin{cases} \mu^m, & -\frac{1}{2} \le \mu < \frac{1}{2} \\ 0, & \text{otherwise} \end{cases} \quad \text{for } N \text{ even and } m \text{ even}. \tag{5.5c}$$

This case is shown in Figure 5.1e.

The third case applies if N is even, $n = 0$, and m is odd. Because the corresponding coefficient b_{m0} is constrained to zero (3.130c), this basis function does not contribute to $h_c(t)$. It is therefore set to zero by definition as represented in Figure 5.1f

$$g(m, 0, \mu) = 0 \quad \text{for } N \text{ even and } m \text{ odd}. \tag{5.5d}$$

In this way, the continuous impulse response of the symmetric piecewise polynomial resampling functions of arbitrary orders can be described using the basis functions (5.5a) to (5.5d).
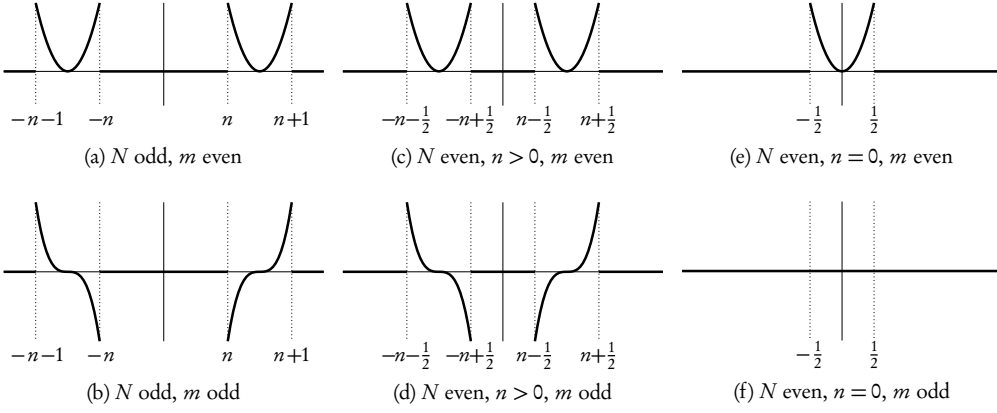
Figure 5.1: Modified basis functions $g(m, n, \mu)$ of symmetric piecewise polynomial resampling functions.

Because the impulse response $h_c(t)$ (5.3) is linear with respect to the basis functions $g(m, n, \mu)$, the continuous frequency response $H_c(j\omega)$ is a linear combination of the continuous Fourier transform of these basis functions.

$$H_c(j\omega) = \mathscr{F}\{h_c(t)\}$$

$$= \sum_{m=0}^{M}\sum_{n=0}^{N'} b_{mn}\mathscr{F}\left\{g\left(m, n, \tfrac{t}{T_i}\right)\right\} \tag{5.6a}$$

$$= T_i \sum_{m=0}^{M}\sum_{n=0}^{N'} b_{mn} G(m, n, \omega) \qquad \text{with} \tag{5.6b}$$

$$G(m, n, \omega) = \mathscr{F}\{g(m, n, \mu)\}. \tag{5.6c}$$

As motivated in Section 3.8, the frequency variable ω is normalized to the input sampling period, that is $\omega = \Omega T_i$ (3.217), for notational convenience.

To obtain the continuous frequency responses $G(m, n, \omega)$, the definition of the continuous Fourier transform (3.7a) is applied. For odd N, the Fourier transform of $g(m, n, \mu)$ (5.5a) is given by

$$G(m, n, \omega) = \int_{-\infty}^{\infty} g(m, n, \mu)e^{-j\omega\mu}\, d\mu \tag{5.7a}$$

$$= \int_{-n-1}^{-n} (\mu + n + \tfrac{1}{2})^m e^{-j\omega\mu}\, d\mu + \int_{n}^{n+1} (-1)^m(\mu - n - \tfrac{1}{2})^m e^{-j\omega\mu}\, d\mu. \tag{5.7b}$$

Substituting the variable μ in the two integrals by

$$\mu_1 = \mu + \left(n + \tfrac{1}{2}\right) \qquad \mu = \mu_1 - \left(n + \tfrac{1}{2}\right) \qquad d\mu_1 = d\mu$$

$$\mu_2 = \mu - \left(n + \tfrac{1}{2}\right) \qquad \mu = \mu_2 + \left(n + \tfrac{1}{2}\right) \qquad \mathrm{d}\mu_2 = \mathrm{d}\mu$$

yields

$$G(m,n,\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu_1^m \, e^{-\mathrm{j}\omega\left(\mu_1 - \left[n+\frac{1}{2}\right]\right)} \, \mathrm{d}\mu_1 + \int_{-\frac{1}{2}}^{\frac{1}{2}} (-1)^m \mu_2^m \, e^{-\mathrm{j}\omega\left(\mu_2 + \left[n+\frac{1}{2}\right]\right)} \, \mathrm{d}\mu_2 \qquad (5.8a)$$

$$= e^{\mathrm{j}\left(n+\frac{1}{2}\right)\omega} \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu_1^m \, e^{-\mathrm{j}\omega\mu_1} \, \mathrm{d}\mu_1 + (-1)^m e^{-\mathrm{j}\left(n+\frac{1}{2}\right)\omega} \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu_2^m \, e^{-\mathrm{j}\omega\mu_2} \, \mathrm{d}\mu_2 . \qquad (5.8b)$$

Because the integral expressions in (5.8b) are equivalent, the variables μ_1 and μ_2 are substituted by μ again. Moving the terms which are independent of the integration variable out of the integrals results in

$$G(m,n,\omega) = \left[e^{\mathrm{j}\left(n+\frac{1}{2}\right)\omega} + (-1)^m e^{-\mathrm{j}\left(n+\frac{1}{2}\right)\omega} \right] \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu^m \, e^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu \qquad (5.9)$$

Applying the relations between exponential and trigonometric functions

$$\cos(x) = \frac{e^{\mathrm{j}x} + e^{-\mathrm{j}x}}{2} \qquad (5.10a)$$

$$\sin(x) = \frac{e^{\mathrm{j}x} - e^{-\mathrm{j}x}}{2\mathrm{j}} , \qquad (5.10b)$$

equation (5.9) can be expressed as a combination of two functions $\Psi(m,n,\omega)$ and $\Phi(m,\omega)$

$$G(m,n,\omega) = \Psi(m,n,\omega)\Phi(m,\omega) \qquad \text{with} \qquad (5.11a)$$

$$\Phi(m,\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu^m \, e^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu \qquad (5.11b)$$

$$\Psi(m,n,\omega) = \begin{cases} 2\cos\left(\left[n+\tfrac{1}{2}\right]\omega\right), & m \text{ even} \\ 2\mathrm{j}\sin\left(\left[n+\tfrac{1}{2}\right]\omega\right), & m \text{ even} \end{cases} \qquad \text{for } N \text{ odd}. \qquad (5.11c)$$

For even N, $n > 0$, the Fourier transform of the basis function $g(m,n,\mu)$ (5.5b) is given by

$$G(m,n,\omega) = \int_{-n-\frac{1}{2}}^{-n+\frac{1}{2}} (\mu+n)^m \, e^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu + \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} (-1)^m (\mu-n)^m \, e^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu . \qquad (5.12a)$$

Using manipulations identical to odd-order case above, $G(m,n,\omega)$ can be written as

$$G(m,n,\omega) = \left[e^{j\omega} + (-1)^m e^{-j\omega} \right] \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu^m e^{-j\omega\mu} \, d\mu \tag{5.12b}$$

$$= \Psi(m,n,\omega)\Phi(m,\omega) \qquad \text{with} \tag{5.12c}$$

$$\Psi(m,n,\omega) = \begin{cases} 2\cos(n\omega), & m \text{ even} \\ 2j\sin(n\omega), & m \text{ odd} \end{cases} \quad for \text{ } N \text{ odd and } n > 0, \tag{5.12d}$$

where the function $\Phi(m,w)$ is identical to (5.11b).

For even N, $n = 0$, m even, the basis function (5.5c) consists of a single polynomial piece centered with respect to $\mu = 0$. Its Fourier transform is given by

$$G(m,0,\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu^m e^{-j\omega\mu} \, d\mu, \tag{5.13}$$

which is expressed in the form of (5.11c) and (5.12d) as

$$G(m,0,\omega) = \Phi(m,\omega)\Psi(m,0,\omega) \qquad \text{with} \tag{5.14a}$$
$$\Psi(m,0,w) = 1 \qquad\qquad\qquad \text{for } N \text{ even}, n = 0 \text{ and } m \text{ even}. \tag{5.14b}$$

Finally, for N even, $n = 0$, m odd,

$$G(m,0,\omega) = 0 \qquad \text{for } N \text{ even}, n = 0 \text{ and } m \text{ odd} \tag{5.15}$$

follows directly from (5.5d).

The different variants of the basis function $G(m,n,\omega)$, (5.11c), (5.12d), (5.14b), and (5.15) can be combined into a single expression

$$G(m,n,\omega) = \Psi(m,n,\omega)\Phi(m,\omega) \quad \text{with} \tag{5.16a}$$

$$\Psi(m,n,\omega) = \begin{cases} 1, & N \text{ even}, n = 0, m \text{ even} \\ 0, & N \text{ even}, n = 0, m \text{ odd} \\ 2\cos(n\omega), & N \text{ even}, n > 0, m \text{ even} \\ 2j\sin(n\omega), & N \text{ even}, n > 0, m \text{ odd} \\ 2\cos\left(\left[n + \frac{1}{2}\right]\omega\right), & N \text{ odd}, m \text{ even} \\ 2j\sin\left(\left[n + \frac{1}{2}\right]\omega\right), & N \text{ odd}, m \text{ odd} \end{cases} \tag{5.16b}$$

Because the scaling function $\Phi(m,\omega)$ is independent of the coefficient index n, $H_c(j\omega)$ (5.6b) can be expressed as

$$H_c(j\omega) = T_i \sum_{m=0}^{M} \sum_{n=0}^{N'} b_{mn} \Phi(m,\omega)\Psi(m,n,\omega)$$

$$= T_i \sum_{m=0}^{M} \Phi(m,\omega) \sum_{n=0}^{N'} b_{mn} \Psi(m,n,\omega). \tag{5.17}$$

By comparing (5.17) to the discrete-time frequency response of the modified Farrow structure (3.126), it becomes apparent that the inner summation represents the non-causal discrete-time frequency response of the subfilters of the Farrow structure, $C_m(\omega)$ (3.126b). Thus, $H_c(\mathrm{j}\omega)$ is compactly represented by

$$H_c(\mathrm{j}\omega) = T_i \sum_{m=0}^{M} \Phi(m,\omega) C_m(\omega). \tag{5.18}$$

Consequently, the continuous frequency response of the modified Farrow structure is a superposition of the discrete-time (and thus periodic) frequency responses of the subfilters $C_m(z)$, each weighted by a non-periodic scaling function $\Phi(m,\omega)$. A similar relation between $H_c(\mathrm{j}\omega)$ and the discrete frequency responses $C_m(\omega)$ has been first pointed out in [Ves00], but this derivation is limited to odd subfilter orders N.

It is observed that $\Psi(m,n,\omega)$ (5.16b) is a real and even function for even m. Conversely, it is odd and purely imaginary if m is odd.

5.2.2 A Closed-Form Expression for the Scaling Function $\Phi(m,\omega)$

To obtain an explicit expression for the scaling function $\Phi(m,\omega)$, the definite integral

$$\Phi(m,\omega) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \mu^m \mathrm{e}^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu \tag{5.19}$$

can be expressed by windowing $\mu \mathrm{e}^{-\mathrm{j}\omega\mu}$ with a unit rectangle (3.174) and integrating the resulting function over the whole real line

$$\Phi(m,\omega) = \int_{-\infty}^{\infty} \Pi(\mu) \mu^m \mathrm{e}^{-\mathrm{j}\omega\mu} \, \mathrm{d}\mu. \tag{5.20}$$

According to the definition of the Fourier integral (3.7a), equation (5.20) is the continuous Fourier transform of the function $\Pi(\mu)\mu^m$

$$\Phi(m,\omega) = \mathscr{F}\{\Pi(\mu)\mu^m\}. \tag{5.21}$$

Utilizing the functional relationship [BSMM06]

$$\mathscr{F}\{t^m f(t)\} = \mathrm{j}^m \frac{\mathrm{d}^m}{\mathrm{d}\omega^m} \mathscr{F}\{f(t)\}, \tag{5.22}$$

which is the dual of the derivative theorem of the continuous Fourier transform [Bra00, Smi10b], and the transform pair for the unit rectangle function (3.175)

$$\mathscr{F}\{\Pi(t)\} = \operatorname{sinc}\left(\frac{\omega}{2}\right),$$

$\Phi(m,\omega)$ is finally expressed in explicit form by

$$\Phi(m,\omega) = j^m \frac{d^m}{d\omega^m} \operatorname{sinc}\left(\frac{\omega}{2}\right). \tag{5.23}$$

This representation makes several properties of the scaling function $\Phi(m,\omega)$ apparent. First, as $\operatorname{sinc}(x)$ is continuous and infinitely differentiable, the functions $\Phi(m,\omega)$ are continuous and are defined on the whole real line for all $m \geq 0$. Second, $\Phi(m,\omega)$ are real-valued functions for m even, while they are purely imaginary for m odd. Third, they are even functions for m even and odd functions if m is odd. That is because $\operatorname{sinc}(x)$ is an even function, and the derivative of an even function is odd, while the derivative of an odd function is even.

Combined with the characteristics of $\Psi(m,n,\omega)$ (5.16b), it follows that the basis functions $G(m,n,w) = \Phi(m,\omega)\Psi(m,n,\omega)$ are real and even irrespective of m and n. This confirms the characterization of $H_c(j\omega)$ as a real and even function, which follows from the real-valued and symmetric nature of $h_c(t)$ [Bra00].

Consequently, the basis functions $G(m,n,\omega)$ can be expressed using real-valued functions, denoted $\Psi_r(m,n,\omega)$ and $\Phi_r(m,\omega)$. This is achieved by transposing the imaginary unit j from $\Psi(m,n,\omega)$ to $\Phi(m,\omega)$ for odd m, resulting in

$$G(m,n,\omega) = \Phi_r(m,\omega)\Psi_r(m,n,\omega) \quad with \tag{5.24a}$$

$$\Phi_r(m,\omega) = (-1)^{\left\lceil \frac{m}{2} \right\rceil} \frac{d^m}{d\omega^m} \operatorname{sinc}\left(\frac{\omega}{2}\right) \tag{5.24b}$$

$$\Psi_r(m,n,\omega) = \begin{cases} 1, & N \text{ even}, n = 0, m \text{ even} \\ 0, & N \text{ even}, n = 0, m \text{ odd} \\ 2\cos(n\omega), & N \text{ even}, n > 0, m \text{ even} \\ 2\sin(n\omega), & N \text{ even}, n > 0, m \text{ odd} \\ 2\cos\left(\left[n+\frac{1}{2}\right]\omega\right), & N \text{ odd}, m \text{ even} \\ 2\sin\left(\left[n+\frac{1}{2}\right]\omega\right), & N \text{ odd}, m \text{ odd} \end{cases} \tag{5.24c}$$

The real-valued functions $\Phi_r(m,\omega)$ and $\Psi_r(m,n,\omega)$ are related to their complex-valued counterparts by

$$\Phi_r(m,\omega) = \begin{cases} \Phi(m,\omega), & m \text{ even} \\ j\Phi(m,\omega), & m \text{ odd} \end{cases} \qquad \Psi_r(m,n,\omega) = \begin{cases} \Psi(m,n,\omega), & m \text{ even} \\ -j\Psi(m,n,\omega), & m \text{ odd} \end{cases}. \tag{5.25}$$

That is, the functions are identical for m even, while they are related by phase shifts of j and $-j$ for odd orders m. The real-valued form is particularly useful for coefficient design methods, because the system can be modeled purely by real quantities. On the downside, the real-valued expressions do not reflect the relation to the discrete-time frequency responses of the Farrow subfilters $C_m(\omega)$, where the constant phase shift of $\pi/2$ forms an essential property of the filters for m odd.

In Figure 5.2, $\Phi_r(m,\omega)$ is depicted for the first four values of m.

(a) $\Phi_r(0,\omega)$

(b) $\Phi_r(1,\omega)$

(c) $\Phi_r(2,\omega)$

(d) $\Phi_r(3,\omega)$
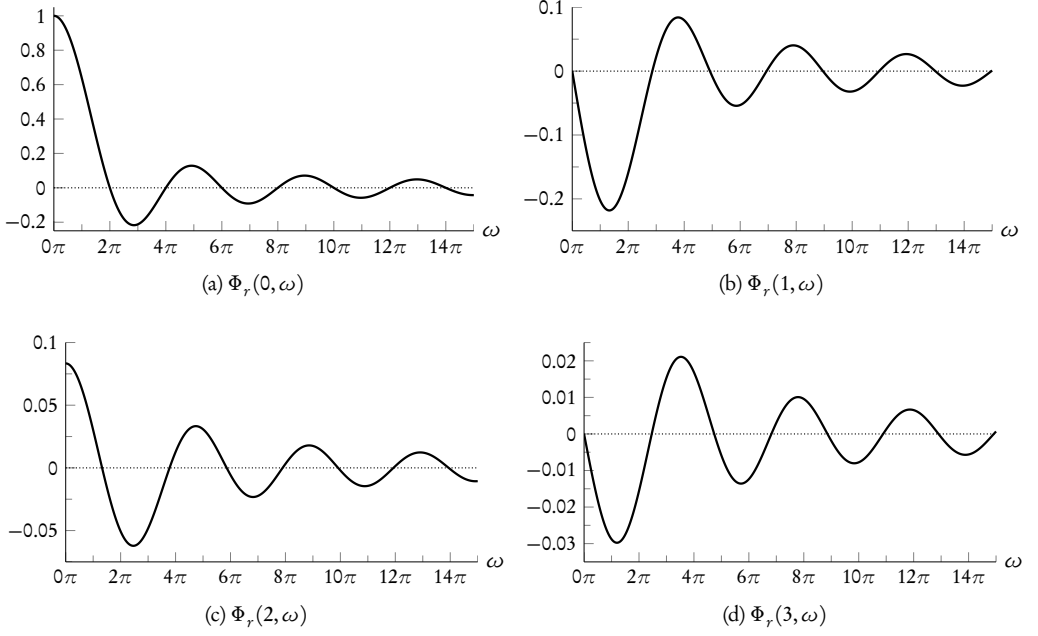
Figure 5.2: Scaling function $\Phi_r(m,\omega)$ of the modified Farrow structure, polynomial orders $m = 0,\ldots,3$.

5.2.3 Stable and Efficient Evaluation of the Scaling Functions $\Phi(m,\omega)$ and $\Phi_r(m,\omega)$

Although (5.23) is a compact, expressive and unambiguous representation of the scaling functions $\Phi(m,\omega)$ and $\Phi_r(m,\omega)$, it cannot be used for direct evaluation due to the derivatives involved. To obtain an explicit formula, the general Leibniz rule [BSMM06, AS65] or Leibniz identity, which generalizes the product rule of differentiation to arbitrary orders

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n}\left[g(x)\cdot h(x)\right] = \sum_{k=0}^{n}\binom{n}{k}\frac{\mathrm{d}^k}{\mathrm{d}x^k}g(x)\cdot\frac{\mathrm{d}^{n-k}}{\mathrm{d}x^{n-k}}h(x), \tag{5.26}$$

is utilized. For this purpose, the function $\mathrm{sinc}(\omega/2)$ is partitioned into factors $g(\omega)$ and $h(\omega)$

$$g(\omega) = \sin\left(\frac{w}{2}\right) \tag{5.27a}$$

$$h(\omega) = 2\omega^{-1}. \tag{5.27b}$$

The derivatives of the factors $g(\omega)$ and $h(\omega)$ are given by

$$\frac{\mathrm{d}^n}{\mathrm{d}\omega^n} g(\omega) = \begin{cases} \frac{1}{2^n} \sin\left(\frac{\omega}{2}\right), & n \bmod 4 = 0 \\ \frac{1}{2^n} \cos\left(\frac{\omega}{2}\right), & n \bmod 4 = 1 \\ -\frac{1}{2^n} \sin\left(\frac{\omega}{2}\right), & n \bmod 4 = 2 \\ -\frac{1}{2^n} \cos\left(\frac{\omega}{2}\right), & n \bmod 4 = 3 \end{cases}$$

$$= \frac{1}{2^n} \sin\left(\frac{\omega}{2} + n\frac{\pi}{2}\right) \tag{5.28a}$$

$$\frac{\mathrm{d}^n}{\mathrm{d}\omega^n} h(\omega) = 2(-1)^n n! w^{-(n+1)} \tag{5.28b}$$

Applying (5.26) to (5.23) yields

$$\Phi(m,\omega) = \mathrm{j}^m 2 \sum_{k=0}^{m} \binom{m}{k} \frac{1}{2^k} \sin\left(\frac{\omega}{2} + k\frac{\pi}{2}\right) \frac{(-1)^{m-k}(m-k)!}{\omega^{m-k+1}}. \tag{5.29}$$

This representation is similar to the weighting function proposed in [Ves00], but the form derived here provides a uniform description for even and odd m, and it is valid for arbitrary subfilters N. In contrast, the expression of [Ves00] is restricted to Farrow structures with odd filter orders.

Although (5.29) is a continuously differentiable function on the whole real line for all m with the exception $\omega = 0$, the summation consists of large values of alternating sign for small absolute values of w. Thus, a numerical evaluation might be subject to instability or increased errors for such arguments. In [VS07], an identical behavior has been observed for the weighting function $\Phi(m, f)$ (3.236) shown in Section 3.8.2. Consequently, a formula based on the power series expansions of the sine and cosine terms (3.238) has been proposed by these authors. However, this formula is still indeterminate for $\omega = 0$ and does not rule out numerical cancellation issues.

In contrast, based on the representation of $\Phi(m, \omega)$ as a derivative (5.23), a compact power series representation is derived here that overcomes these deficiencies. Starting from the power series representation of the sine function (e.g. [BSMM06, Kno96])

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots$$

$$= \sum_{k=0}^{\infty} a_k x^n \quad \text{with} \quad a_k = \begin{cases} \frac{(-1)^{\frac{k+1}{2}}}{k!}, & k \text{ odd} \\ 0, & k \text{ even} \end{cases},$$

the power series representation of $\Phi(0, \omega)$ results in

$$\Phi(0, \omega) = \mathrm{sinc}\left(\frac{\omega}{2}\right) = 1 - \frac{w^2}{24} + \frac{w^4}{1920} - \frac{w^6}{322560} + \frac{w^8}{92897280} - \cdots \tag{5.30a}$$

$$= \sum_{k=0}^{\infty} a_k \omega^k \quad \text{with} \quad a_k = \begin{cases} \frac{(-1)^{\frac{k}{2}}}{(k+1)!2^k}, & k \text{ even} \\ 0, & m \text{ odd} \end{cases}. \tag{5.30b}$$

Generalizing the derivative of a power series [Kno96]

$$\frac{\mathrm{d}}{\mathrm{d}x} f(x) = \sum_{k=0}^{\infty} a_{k+1} x^k$$

$$= \sum_{k=0}^{\infty} a'_k x^k \qquad \text{with} \quad a'_k = (k+1) a_{k+1}$$

to derivatives of arbitrary orders yields

$$\frac{\mathrm{d}^n}{\mathrm{d}x^n} \sum_{k=0}^{\infty} a_k x^k = \sum_{k=0}^{\infty} a_{k+n} (k+n)! x^k \tag{5.31a}$$

$$= \sum_{k=0}^{\infty} a_k^{(n)} x^k \qquad \text{with} \quad a_k^{(n)} = a_{k+n} (k+1)^{\overline{n}}, \tag{5.31b}$$

where $x^{\overline{n}}$ denotes the rising factorial [GKP06]

$$x^{\overline{n}} = \underbrace{x(x+1)\cdots(x+n-1)}_{n} = \frac{(x+n-1)!}{(x-1)!}. \tag{5.32}$$

Application of (5.31) to the power series expansion of $\Phi(0,\omega)$ (5.30) yields

$$\Phi(m,\omega) = \mathrm{j}^m \sum_{k=0}^{\infty} a_k^{(m)} \omega^k \quad \text{with} \tag{5.33a}$$

$$a_k^{(m)} = \begin{cases} \frac{(-1)^{\frac{k+m}{2}} (k+1)^{\overline{m}}}{2^{k+m}(k+m+1)!}, & k+m \text{ even} \\ 0, & k+m \text{ odd} \end{cases} \tag{5.33b}$$

The power series expansion (5.33) differs from the expression (3.238) given in [VS07] in several aspects: First, because (3.238) is based on the power series expansions of the individual sine and cosine components, it utilizes a nested summation that is more prone to numerical cancellation than the single summation used in the proposed formula. Second, (3.238) contains negative powers of the frequency variable, thus making the formula indeterminate at $\omega = 0$. Likewise, small absolute values of ω cause large terms of alternating sign, which again bears the potential for numerical cancellation. In contrast, the proposed equation contains only positive powers of ω. Thus the resulting function is smooth on the whole real line, while the addends of (5.33) are convergent for $\omega \to 0$. Finally, the series expansion (5.33) provides a convenient way to determine the value $\Phi(m,0)$:

$$\Phi(m,0) = \mathrm{j}^m c_0^{(m)} = \mathrm{j}^m \begin{cases} \frac{(-1)^{\frac{m}{2}}}{2^m (m+1)!} 1^{\overline{m}}, & m \text{ even} \\ 0, & m \text{ odd} \end{cases} \tag{5.34a}$$

$$= \begin{cases} \frac{1}{2^m (m+1)}, & m \text{ even} \\ 0, & m \text{ odd} \end{cases} . \tag{5.34b}$$

m	0	1	2	3	4	5	6	7	8	9	10
$\Phi(m,0)$	1	0	$\frac{1}{12}$	0	$\frac{1}{80}$	0	$\frac{1}{448}$	0	$\frac{1}{2304}$	0	$\frac{1}{11264}$

Table 5.1: Values of the scaling function $\Phi(m,\omega)$ for $\omega = 0$, $m = 0, 1, \ldots, 10$.

In Table 5.1, $\Phi(m,0)$ is shown for some small values of m. It is noted that these values are identical for the real-valued variant $\Phi_r(m,0)$ due to the identity (5.25), because $\Phi(m,0)$ is nonzero only for even m.

5.2.4 Application to Design Methods

As expressed by (5.17), the continuous frequency response of the modified Farrow structure is linear with respect to the elements b_{mn} of the coefficient matrix B. Therefore, optimization methods such as linear programming or weighted least squares minimization can be applied directly, similar to their use in discrete-time FIR filter design (e.g. [Sar93, OSB99]). However, due to the continuous-time nature of $H_c(\mathrm{j}\omega)$, the approximation region X is of infinite extent as described in Section 3.8. This distinction affects design methods for ASRC in two ways.

First, for practical coefficient design methods, a finite upper bound ω_{up} for the stopband approximation region has to be determined. A bound that is chosen larger than necessary only results in an increased design time and potential numerical errors due to the increased number of grid points. In contrast, if the bound is set to a too small value, the designed system exhibits significant error components above ω_{up}, thus degrading the validity of the design. Nevertheless, in case of the modified Farrow structure, the selection of an appropriate approximation region is straightforward in practice. Empirical tests suggest that the required upper bound ω_{up} depends primarily on the polynomial order M and increases monotonically with respect to this parameter. In any way, in accordance with the general design process for digital filters (e.g. [Sar93]), the obtained filter coefficients should be validated on a different, typically larger and denser frequency grid. In this way, an improper upper bound is detected and corrected without difficulty.

The discretization of the frequency variable forms the second distinction to discrete-time filter design. In the latter field, several design methods that do not require an explicit discretization have been proposed, including the Parks-McClellan algorithm [PM72b, PM72a, OSB99, KM99], optimization based on semi-infinite programming [Pot97] or linear matrix inequality formulations [DLS02, Dav10]. However, these design methods are generally tightly bound to the representation of the filter to be optimized, in particular the basis functions and the approximation region used. For these reasons, they cannot be adapted to the design of ASRC systems straightforwardly. On the other hand, design methods based on discrete frequency grids are widely used, e.g. in FIR filter design methods based on linear programming [Rab72, Sar93]. At the same time, advances in numerical optimization allow the use of very dense discretization grids without excessively increased design times. For this reason, methods based on an explicit discretization of the approximation region are used exclusively in the following. This choice is in accordance with the design methods for the modified Farrow structure proposed in [Ves99, VS07].

In this way, the approximation region X is represented by $\mathbf{X}$, a set of $K + 1$ discrete frequencies

$$\mathbf{X} = \{\omega_1, \omega_1, \ldots, \omega_K\} = X_p \cup X_s \qquad \text{with} \tag{5.35a}$$

$$\mathbf{X}_p = \{\omega_{p0}, \omega_{p1}, \ldots, \omega_{pK_p}\} \tag{5.35b}$$

$$\mathbf{X}_s = \{\omega_{s0}, \omega_{s1}, \ldots, \omega_{sK_s}\} \tag{5.35c}$$

$$K = K_p + K_s + 1, \tag{5.35d}$$

comprising the discretized passband and stopband regions $\mathbf{X}_p$ and $\mathbf{X}_s$. These grids consist of $K_p + 1$ and $K_s + 1$ discrete frequencies, respectively. The discretized stopband grid $\mathbf{X}_s$ may approximate uniform (3.225) as well as nonuniform stopband specifications (3.225).

The discretized continuous frequency response $\mathbf{H}_c$ is a vector of size $K + 1$

$$\mathbf{H}_c[k] = H_c\left(j\omega_k\right) \quad \text{for } k = 0, 1, \ldots, K \tag{5.36}$$

and is determined by the matrix equation

$$\mathbf{H}_c = \mathbf{G} \cdot \mathbf{b} \tag{5.37}$$

comprising the discrete transfer matrix $\mathbf{G}$ and the coefficient vector $\mathbf{b}$. The vector $\mathbf{b}$ is formed by transposing and concatenating the rows of the coefficient vector B (3.129)

$$\mathbf{b} = \begin{bmatrix} b_{00} & \cdots & b_{0N'} & b_{10} & \cdots & b_{1N'} & \cdots & b_{M0} & \cdots & b_{MN'} \end{bmatrix}'. \tag{5.38}$$

For even orders N, the zero-valued elements resulting from the condition (3.130c) are excluded. Thus, the size of $\mathbf{b}$ is determined by

$$N_{coeffs} = \begin{cases} \frac{(N+1)}{2}(M+1), & N \text{ odd} \\ \frac{N}{2}(M+1) + \left\lceil \frac{M+1}{2} \right\rceil, & N \text{ even} \end{cases}. \tag{5.39}$$

The discretized transfer matrix $\mathbf{G}$ is of dimension $(K+1) \times N_{coeffs}$

$$\mathbf{G} = \begin{bmatrix} G(0,0,\omega_0) & \cdots & G(0,N',\omega_0) & \cdots & G(M,0,\omega_0) & \cdots & G(M,N',\omega_0) \\ G(0,0,\omega_1) & \cdots & G(0,N',\omega_1) & \cdots & G(M,0,\omega_1) & \cdots & G(M,N',\omega_1) \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ G(0,0,\omega_K) & \cdots & G(0,N',\omega_K) & \cdots & G(M,0,\omega_K) & \cdots & G(M,N',\omega_K) \end{bmatrix}. \tag{5.40}$$

Each row k represents the frequency response for a discrete frequency ω_k, while column l contains the transfer functions $G(m,n,\omega_k)$ (5.16) corresponding to the l-th element of the coefficient vector $\mathbf{b}[l] = b_{mn}$. In discretized form, the desired frequency response $\hat{H}_c(j\omega)$ is represented by

$$\hat{\mathbf{H}}_c[k] = \hat{H}_c(j\omega_k) = \begin{cases} 1, & \omega_k \in X_p \\ 0, & \omega_k \in X_s \end{cases} \qquad \text{for } k = 0, 1, \ldots, K. \tag{5.41}$$

As argued in Section 3.8.1, the magnitude T_i of the ideal anti-imaging/anti-aliasing filter (3.215) is omitted here deliberately, corresponding to a normalization of the input sampling period to

(a) Extended rectangle rule, equidistant grid   (b) Extended trapezoidal rule, arbitrary grid
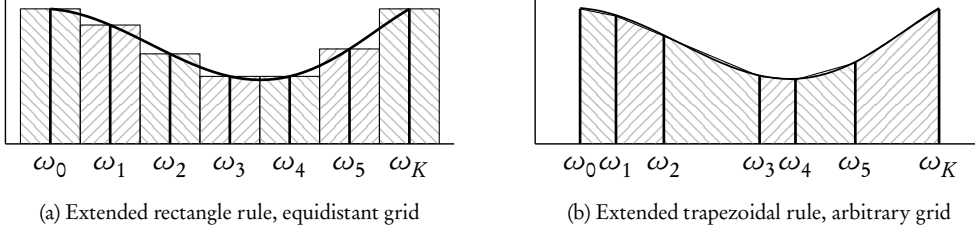
Figure 5.3: Numerical integration schemes to approximate the integral weighted squared error for weighted least squares design.

$T_i = 1$. The weighting vector $\mathbf{W}$ is formed by sampling the weighting function $W(\omega)$ at the grid frequencies ω_k

$$\mathbf{W}[k] = W(\omega_k) \qquad \text{for } k = 0, 1, \ldots, K. \tag{5.42}$$

The frequency response (5.37) can be used in standard optimization methods in a straightforward way. For weighted least squares design, as introduced in Section 3.8.2, discretization yields the objective function

$$\operatorname*{minimize}_{\mathbf{b}} \sum_{k=0}^{K} \left(\mathbf{W}[k]\mathbf{s}[k] \left| \mathbf{H}_c[k] - \widehat{\mathbf{H}}_c[k] \right| \right)^2. \tag{5.43}$$

Here, the vector $\mathbf{s}[k]$ denotes an additional weighting to incorporate the influence of the discretization grid. In several approaches to WLS design, e.g. [Sar93, VS07], such a weighting is not used, resulting in two principal effects. First, the sum in (5.43) does not approximate the weighted squared error E_2 (3.226a), but a number proportional to this measure that depends on the number of grid points. Second, this choice effectively implies equidistant grids. The use of dedicated weightings enables to overcome these problems. A constant weighting for equidistant grids

$$\mathbf{s}[k] = \frac{1}{h} \qquad \text{with} \quad h = \omega_k - \omega_{k-1} \tag{5.44}$$

corresponds to an approximation of the integral (3.226a) by numerical integration with an extended or composite rectangle rule as shown in Figure 5.3a. More sophisticated weighting vectors $\mathbf{s}[k]$ enable non-equidistant grids as well as more accurate approximations of the integral. For instance,

$$\mathbf{s}[k] = \begin{cases} \frac{1}{2}(\omega_0 + \omega_1), & k = 0 \\ \frac{1}{2}(\omega_{k+1} - \omega_{k-1}), & 1 \leq k \leq K - 1 \\ \frac{1}{2}(\omega_{K-1} + \omega_K), & k = K \end{cases} \tag{5.45}$$

represents the use of the extended trapezoidal rule with non-uniform intervals [PTVF92, Sch97], thus facilitating arbitrary discretization grids. This case is depicted in Figure 5.3b.

Using this representation, the solution to the optimization problem (5.43) is given by

$$\mathbf{b} = \left(\mathbf{G}_W^T \mathbf{G}_W \right)^{-1} \mathbf{G}_W \widehat{\mathbf{H}}_W \quad \text{with} \tag{5.46a}$$

$$\mathbf{G}_W = \begin{bmatrix} \mathbf{W}[0]\mathbf{s}[0]\mathbf{G}[0,0] & \mathbf{W}[0]\mathbf{s}[0]\mathbf{G}[0,1] & \cdots & \mathbf{W}[0]\mathbf{s}[0]\mathbf{G}[0,N'] \\ \mathbf{W}[1]\mathbf{s}[1]\mathbf{G}[1,0] & \mathbf{W}[1]\mathbf{s}[1]\mathbf{G}[1,1] & \cdots & \mathbf{W}[1]\mathbf{s}[1]\mathbf{G}[1,N'] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}[K]\mathbf{s}[K]\mathbf{G}[0,K] & \mathbf{W}[K]\mathbf{s}[K]\mathbf{G}[K,1] & \cdots & \mathbf{W}[K]\mathbf{s}[K]\mathbf{G}[K,N'] \end{bmatrix} \tag{5.46b}$$

$$\widehat{\mathbf{H}}_W = \begin{bmatrix} \mathbf{W}[0]\mathbf{s}[0]\widehat{\mathbf{H}}_c[0] & \mathbf{W}[1]\mathbf{s}[1]\widehat{\mathbf{H}}_c[1] & \cdots & \mathbf{W}[K]\mathbf{s}[K]\widehat{\mathbf{H}}_c[K] \end{bmatrix}, \tag{5.46c}$$

where $\mathbf{G}_W$ denotes the transfer matrix $\mathbf{G}$ scaled by the weighting function $W(\omega)$ and the integration weights $\mathbf{s}[k]$ for the discrete frequencies $\omega_k \in \mathbf{X}$.

For weighted Chebyshev (minimax) approximation, the discretized design objective is

$$\underset{\mathbf{b}}{\text{minimize}} \; \delta_\infty = \left\| \mathbf{W}[k] \left(\mathbf{G}[k]\mathbf{b}[k] - \widehat{\mathbf{H}}_c[k] \right) \right\|_\infty, \tag{5.47}$$

where $\|\mathbf{x}\|_\infty$ denotes the maximum or infinity vector norm

$$\|\mathbf{x}\|_\infty = \max\left(|x_0|, |x_1|, \ldots, |x_N| \right). \tag{5.48}$$

For filter design problems, the objective function is typically transformed into the standard form of linear programming [Rab72, Sar93]

$$\underset{\mathbf{b}}{\text{minimize}} \; \delta_\infty \quad \text{subject to} \quad \mathbf{Gb} - \frac{1}{\mathbf{W}}\delta \leq \widehat{\mathbf{H}}_c \tag{5.49a}$$

$$-\mathbf{Gb} - \frac{1}{\mathbf{W}}\delta \leq -\widehat{\mathbf{H}}_c, \tag{5.49b}$$

where $1/\mathbf{W}$ denotes the piecewise reciprocal of $\mathbf{W}$. The problem (5.49) can be solved using standard methods for linear programming.

Disciplined Convex Programming   Least squares and linear programming problems are subsets of convex optimization problems [BV04], a wide class of mathematical optimization problems that can be stated in the form

$$\underset{x}{\text{minimize}} \; f(x) \tag{5.50a}$$

$$\text{subject to } g_i(x) \leq 0 \qquad\qquad i = 0, 1, \ldots, n_g \tag{5.50b}$$

$$h_j(x) = 0 \qquad\qquad j = 0, 1, \ldots, n_h. \tag{5.50c}$$

Here, $f(x)$ is the objective function to be minimized, and $g_i(x)$ and $h_j(x)$ denote potentially empty sets of inequality and equality constraints, respectively. $f(x)$ and $g_i(x)$, $i = 0, 1, \ldots, n_g$ are required to be convex functions, while the equality constraints $h_j(x)$, $j = 0, 1, \ldots, n_h$ must be affine. The formulation as a convex optimization problem enables a uniform representation of the ASRC design problems (5.43) and (5.49). At the same time, it allows the use of sophisticated numerical methods for convex optimization which have progressed significantly in the recent years.

A particularly useful methodology for modeling and solving convex optimization problems has been introduced as *disciplined convex programming* [Gra04, GB08]. Instead of forcing the user

to specify or transform the optimization problem into a standard form for convex optimization, the problem can be stated in an intuitive way using sets of predefined and extensible atoms and rules. The framework of disciplined convex programming checks the problem for adherence to convexity and affinity constraints. It performs transformations into a form suitable for standard numerical convex optimization methods, and solves the problem by calling an appropriate solution algorithm. Implementations of disciplined convex programming are available, for instance CVXMOD[1] for use with the Python programming language or CVX[2] for MATLAB.

As an example, the weighted least-squares design specification (5.46a) is formulated in CVX as follows:

```
cvx_begin
    variable b( N_coeffs )
    minimize ( norm( W . (G b − Ĥ_c ), 2 ));
cvx_end
```

Here, **norm**$(\cdot, 2)$ denotes the L_2-norm. In contrast, the minimax objective function (5.47) translates to

```
cvx_begin
    variable b( N_coeffs )
    minimize ( max( abs( W . (G b − Ĥ_c ))));
cvx_end
```

or, by directly stating the L_∞ norm as **norm**$(\cdot, \textbf{Inf})$

```
cvx_begin
    variable b( N_coeffs )
    minimize ( norm( W . (G b − Ĥ_c ), Inf ));
cvx_end
```

These examples show that standard design objectives can be stated in the disciplined convex programming framework in a compact and expressive manner. However, the main advantage of this framework is the ability to include additional conditions and constraints in an extensible and intuitive way. For instance, if the worst-case stopband error shall be minimized with respect to a fixed maximum passband error δ_p, the resulting CVX program is formulated as

```
cvx_begin
    variable b(N_coeffs)
    minimize ( norm( W(X_s) . (G(X_s) b − Ĥ_c(X_s) ), Inf ));
    subject to abs( G(X_p) b − Ĥ_c(X_p)) <= δ_p ones(K_p+1,1);
cvx_end
```

where $\mathbf{W}(\mathbf{X}_p)$, $\mathbf{G}(\mathbf{X}_s)$, … represent the partition of the given vector or matrix corresponding to the passband grid $\mathbf{X}_p$ or stopband grid $\mathbf{X}_s$, respectively. If the stopband error shall be minimized with respect to a weighted least squares criterion subject to a fixed maximum passband error,

---

[1] http://cvxmod.net, retrieved 2011-03-21
[2] http://cvxr.com/cvx/, retrieved 2011-03-21

changing the norm in the above program to **norm**$(\cdot, 2)$ is sufficient. Thus, design objectives in the style of peak-constrained least square filter design [SLB96, AS98, Dav10] are easily achieved.

As stated in Section 3.8, many design specifications that are expressible as conditions in the time or the frequency domain are frequently used in ASRC design. For example, this includes the interpolation condition and continuity or regularity constraints. Such constraints typically decrease the degree of freedom of the design. With disciplined convex programming, such constraints can be added as equality conditions and take the form

$$\mathbf{C}_{eq}\mathbf{b} = \mathbf{c}_{eq}, \tag{5.51}$$

where $\mathbf{C}_{eq}$ is a matrix of dimension $(n_j \times N_{coeffs})$ and $\mathbf{c}_{eq}$ is a vector of length n_j forming the right hand side of the linear system. As in (5.50), n_j is the number of scalar equality constraints.

The Interpolation Condition   The linear constraints used to specify the interpolation condition (3.232) differ between even and odd subfilter orders. For N even, fixed values are explicitly determined for a subset of coefficients

$$b_{0n} = \begin{cases} 1, & n = 0 \\ 0, & n = 1,\ldots,N' \end{cases}. \tag{5.52}$$

For odd N, the interpolation condition results in a system of linear equations

$$\sum_{m=0}^{M} b_{m0}\left(-\frac{1}{2}\right)^m = 1 \tag{5.53a}$$

$$\sum_{m=0}^{M} b_{mn}\left(\frac{1}{2}\right)^m = 0 \quad \text{for } n = 0, 1 \ldots N'. \tag{5.53b}$$

Moreover, for even orders N, the interpolation condition requires $h_c(t)$ to be continuous.

Continuity   The continuity of the continuous-time impulse response can be formulated as a set of linear equality constraints that restrict the values of the polynomial pieces $h(n, \mu)$ (3.98) of $h_c(t)$ at the junction points

$$h(0, \mu_{min}) = 0 \tag{5.54a}$$
$$h(n, \mu_{max}) = h(n+1, \mu_{min}) \qquad \text{for } n = 0, 1, \ldots, N-1 \tag{5.54b}$$
$$h(N, \mu_{max}) = 0 \tag{5.54c}$$

For symmetric piecewise polynomial basis functions, these conditions can be specified as a linear system utilizing the definition of the modified basis functions $g(m, n, \mu)$ of the impulse response (5.3)

$$\sum_{m=0}^{M} b_{m0}\left(\frac{1}{2}\right)^m - \sum_{m=0}^{M} b_{m0}\left(-\frac{1}{2}\right)^m = 0 \qquad \text{for } N \text{ even} \tag{5.55a}$$

$$\sum_{m=0}^{M} b_{mn}\left(\frac{1}{2}\right)^m - \sum_{m=0}^{M} b_{m(n+1)}\left(-\frac{1}{2}\right)^m = 0 \qquad \text{for } n = 0, 1, \ldots, N'-1 \tag{5.55b}$$

$$\sum_{m=0}^{M} b_{mN'} \left(\frac{1}{2}\right)^m = 0. \tag{5.55c}$$

Regularity   The condition of regularity, that is the number of continuous derivatives of $h_c(t)$, can be considered as an extension of continuity. Continuity corresponds to regularity of order 0, while regularity of order k implies regularity of all orders less than k, including continuity. In terms of $h_c(t)$, this condition is expressed by

$$\frac{\mathrm{d}^l}{\mathrm{d}\mu^l} h(0, \mu_{min}) = 0 \qquad\qquad \text{for } l = 0, 1, \dots, k \tag{5.56a}$$

$$\frac{\mathrm{d}^l}{\mathrm{d}\mu^l} h(n, \mu_{max}) = \frac{\mathrm{d}^l}{\mathrm{d}\mu^l} h(n+1, \mu_{min}) \quad \text{for } n = 0, 1, \dots, N-1 \text{ and } l = 0, 1, \dots, k \tag{5.56b}$$

$$\frac{\mathrm{d}^l}{\mathrm{d}\mu^l} h(N, \mu_{max}) = 0 \qquad\qquad \text{for } l = 0, 1, \dots, k \tag{5.56c}$$

Applied to the parametric representation of symmetric piecewise polynomial basis functions, these conditions are stated as

$$\sum_{m=0}^{M-l} m^{\overline{l}} b_{(m+l)n} \left(\frac{1}{2}\right)^m - \sum_{m=0}^{M} m^{\overline{l}} b_{(m+l)(n+1)} \left(-\frac{1}{2}\right)^m = 0 \quad \text{for } n = 0, 1, \dots, N'-1 \tag{5.57a}$$

$$\sum_{m=0}^{M} m^{\overline{l}} b_{mN'} \left(\frac{1}{2}\right)^m = 0 \tag{5.57b}$$

and, additionally,

$$\sum_{m=0}^{M-l} m^{\overline{l}} b_{m0} \left(\frac{1}{2}\right)^m - \sum_{m=0}^{M} m^{\overline{l}} b_{m0} \left(-\frac{1}{2}\right)^m = 0 \quad \text{for } N \text{ even}. \tag{5.57c}$$

These equality conditions must hold for all orders $l = 0, 1, \dots, k$, with $x^{\overline{l}}$ denoting the rising factorial defined in (5.32).

The interpolation condition, continuity and regularity constraints are readily transformed and combined into a linear equation system in matrix form according to (5.51). This constraint can be directly incorporated in a CVX program.

```
cvx_begin
    variable b( N_coeffs );
    minimize( norm( W(X_s) . (G(X_s) b - Ĥ_c)(X_s) ), Inf ));
    subject to abs( G(X_p) b - Ĥ_c(X_p)) <= δ_p ones(K_p+1,1);
    subject to C_eq b == c_eq;
cvx_end
```

(a) Frequency response

(b) Passband detail

(c) Stopband detail
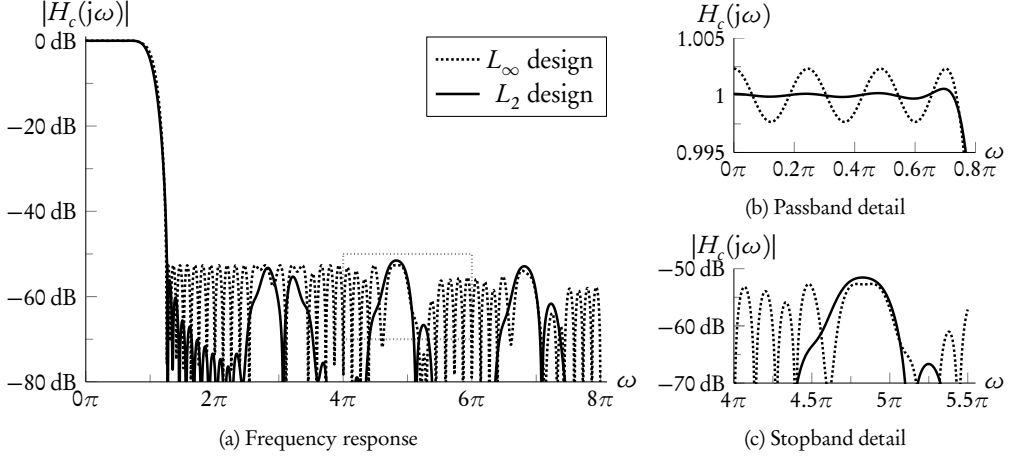
Figure 5.4: Continuous-time frequency response of the modified Farrow structure designed according to the L_2 and L_∞ norms. Orders $M = 3$ and $N = 15$, cutoff frequency $\omega_c = 0.75\pi$, uniform error weighting, uniform stopbands.

In [VS07], a fixed set of constraints, namely the interpolation condition, impulse response continuity and a combination of these constraints, have been implemented for odd orders N. For each constraint, this implementation performs a specific transformation on the coefficient set $\{b_{mn}\}$, reducing the number of free variables to be optimized. In contrast, the approach presented here does not reduce the dimension of the design problem. However, the intuitive specification of the conditions and the ability to combine arbitrary sets of constraints with a selectable error norm results in a very flexible design method.

To illustrate the design framework for the modified Farrow structure, the continuous frequency responses for an ASRC system with orders $M = 3$, $N = 15$ and cutoff frequency $\omega_c = 0.75\pi$ designed with respect to the L_2 and L_∞ norms is shown in Figure 5.4. For both designs, a constant weighting function $W(\omega) = 1$ and an upper limit for the optimization grid $\omega_{up} = 7\pi$ have been used. The L_∞ design exhibits approximately equiripple behavior in the stopband up to about 4π and is dampened towards higher frequencies. The stopband error of the system designed in the L_2 sense is considerably lower than in case of the L_∞ design for most of $\mathbf{X}_s$ and overshoots the stopband error of the latter design only in the region around 5π, that is, the second image of the passband.

For the filter orders and conditions considered in [VS07], the obtained coefficient sets as well as the resulting filter quality are very similar to the designs gained by the proposed approach. This supports the conjecture that the optimum design quality for the modified Farrow structure is limited by intrinsic characteristics of this structure rather than by the use of a particular basis function or optimization method. This supposition is further examined in Chapter 6, which also presents a variety of design examples.

5.2.5 Conclusions

Compared to existing approaches, the proposed representation of the continuous frequency response of symmetric piecewise polynomial resampling filters holds for arbitrary orders and has a concise structure that enables stable and efficient evaluation. Even more important, it enables analytical insight into the characteristics of this class of functions, which are efficiently implemented by the modified Farrow structure. In particular, the relation to the discrete-time frequency responses of the subfilters $C_m(z)$ and the role of the scaling functions $\Phi(m, \omega)$ are analyzed. Built upon the descriptions developed in this section, a design framework based on the disciplined convex programming methodology is proposed. It enables flexible, optimal design of the parameters of the modified Farrow structure with respect to a variety of design objectives and constraints.

5.3 A Closed Form for the Continuous Frequency Response of Lagrange Interpolators

5.3.1 Introduction

Lagrange interpolation, as characterized in Section 3.6.3, is widely used in arbitrary sample rate conversion. Despite this widespread application, analytical expressions for the system response are rarely found in the literature on SRC. In most cases, the characterization is limited to qualitative statements of the form "the zeros of the system function tend to be clustered about those frequencies [integer multiples of $2\pi T_i$]" [SR73, Hen02]. Quantitative statements are restricted to the lowest interpolation orders. For instance, a Lagrange interpolator of order $N = 0$ corresponds to a zero-order hold element. Its frequency response, denoted $H_c^{(0)}(\mathrm{j}\omega)$ here, is given by

$$H_c^{(0)}(\mathrm{j}\omega) = \frac{\sin\frac{\omega}{2}}{\frac{\omega}{2}} = \mathrm{sinc}\left(\frac{\omega}{2}\right) \tag{5.58}$$

for instance, in [AK93, Har97, OSB99].

For linear interpolation, that is Lagrange interpolation of order $N = 1$, the frequency response $H_c^{(1)}(\mathrm{j}\omega)$ is derived, e.g., in [EGH93, Har97, Smi10a]

$$H_c^{(1)}(\mathrm{j}\omega) = \mathrm{sinc}\left(\frac{\omega}{2}\right)^2. \tag{5.59}$$

In this section, a concise expression for $H_c^{(N)}(\mathrm{j}\omega)$ of arbitrary orders N is presented that enables efficient and numerically stable evaluation as well as insight into the characteristics of Lagrange interpolators. This derivation is mainly based on [FB09a].

5.3.2 Lagrange Interpolation as Symmetric Polynomial-Based Resampling Filters

As described in Section 3.6.3.1, the filter coefficients $h(n, \mu)$ of a Lagrange interpolator are formed by the Lagrange polynomials, that is, polynomials of order N. Therefore, Lagrange interpolators are contained in the class of polynomial-based resampling filters.

Moreover, if the basepoint set $\{t_n\}$ (3.82), and thus the intersample position range, is chosen symmetric with respect to zero, the coefficient matrix C of the polynomial-based interpolation filter exhibits the coefficient symmetries (3.120). Consequently, the continuous impulse response $h_c(t)$ is symmetric with respect to zero. In other words, the resulting resampling filter can be implemented by the modified Farrow structure as described in Section 3.6.2. It is important to note that the restriction to symmetric basepoint sets does not limit the generality of the frequency response derived here. On the contrary, the system response of Lagrange interpolation is independent of a particular basepoint set, but depends only on the relative position of the intersample position variable μ with respect to the basepoints. The symmetric basepoint set utilized here shows some advantages in the derivation of $H_c^{(N)}(j\omega)$. Nonetheless, the resulting frequency response holds for Lagrange interpolators irrespective of the basepoint set used, as long as the intersample position variable falls into the optimal range as defined by (3.84).

5.3.3 Derivation

For Lagrange interpolation, the Farrow coefficient matrix C is defined by (3.153), where the basepoints t_n are chosen according to the definition of the symmetric basepoint set (3.89). Applying these coefficients to the continuous frequency response of a symmetric piecewise polynomial resampling function (5.18) yields an analytical representation for $H_c^{(N)}(j\omega)$. However, the expressiveness of this formula is severely limited, as the number of addends grows quadratically with respect to the interpolation order N. Such a representation is given, for instance, in [Eva00b].

Evaluating and simplifying $H_c^{(N)}(j\omega)$ for some low orders N reveals that the frequency responses can be represented in a concise form that exhibits a characteristic structure. In Table 5.2, $H_c^{(N)}(j\omega)$ is shown for $N = 0, 1, \ldots, 10$.

It appears that the general structure underlying $H_c^{(N)}(j\omega)$ has the form

$$H_c^{(N)}(j\omega) = \text{sinc}\left(\frac{\omega}{2}\right)^{N+1}\left(1 + \sum_{i=1}^{\left\lfloor\frac{N}{2}\right\rfloor} c_i\,\omega^{2i}\right), \tag{5.60}$$

where $\text{sinc}(x)$ denotes the unnormalized cardinal sine function as defined in (3.29), while the coefficients c_i depend only on the order N. It is noted that the magnitude scaling of T_i in (5.18) is omitted here as motivated in Section 3.8.1, corresponding to a normalization of the input sampling period with $T_i = 1$.

It is most notable that this structure, as well as the formulas for $N = 0, \ldots, 3$, has been proposed by I. J. Schoenberg in the landmark series of papers that introduced spline functions [Sch46a, Sch46b].

In Section 3.6.4.2, the function

$$B^N(\omega) = \text{sinc}\left(\frac{\omega}{2}\right)^{N+1}$$
$$= \mathscr{F}\left\{\beta^N(t)\right\}$$

has been introduced as the Fourier transform of a spline basis function $\beta^N(t)$ of order N. This confirms a strong interrelation between Lagrange interpolation and splines, which has been pointed out by several authors, e.g. [Sch46a, BTU01].

N	$H_c^{(N)}(j\omega)$
0	$\operatorname{sinc}\left(\frac{\omega}{2}\right)$
1	$\operatorname{sinc}\left(\frac{\omega}{2}\right)^2$
2	$\left(1+\frac{1}{8}\omega^2\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^3$
3	$\left(1+\frac{1}{6}\omega^2\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^4$
4	$\left(1+\frac{5}{24}\omega^2+\frac{3}{128}\omega^4\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^5$
5	$\left(1+\frac{1}{4}\omega^2+\frac{1}{30}\omega^4\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^6$
6	$\left(1+\frac{7}{24}\omega^2+\frac{259}{5760}\omega^4+\frac{5}{1024}\omega^6\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^7$
7	$\left(1+\frac{1}{3}\omega^2+\frac{7}{120}\omega^4+\frac{1}{140}\omega^6\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^8$
8	$\left(1+\frac{3}{8}\omega^2+\frac{47}{640}\omega^4+\frac{3229}{322560}\omega^6+\frac{35}{32768}\omega^8\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^9$
9	$\left(1+\frac{5}{12}\omega^2+\frac{13}{144}\omega^4+\frac{41}{3024}\omega^6+\frac{1}{630}\omega^8\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^{10}$
10	$\left(1+\frac{11}{24}\omega^2+\frac{209}{1920}\omega^4+\frac{17281}{967680}\omega^6+\frac{117469}{51609600}\omega^8+\frac{63}{262144}\omega^{10}\right)\operatorname{sinc}\left(\frac{\omega}{2}\right)^{11}$

Table 5.2: Continuous frequency responses of Lagrange interpolators for interpolation orders $N = 0, 1, \ldots, 10$.

The conjectured form of the continuous frequency response $H_c^{(N)}(j\omega)$ (5.60) implies the continuous impulse response

$$h_c^{(N)}(t) = \beta^N(t) + \sum_{i=1}^{\lfloor\frac{N}{2}\rfloor}(-1)^i c_i \frac{d^{2i}}{dt^{2i}}\beta^N(t), \tag{5.61}$$

which is obtained using the Fourier transform pair for time-domain derivatives (5.22). That is, the conjectured impulse response $h_c^{(N)}(t)$ is a linear combination of a spline basis function of order N and its even-order derivatives. The derivatives of $\beta^N(t)$ are conveniently obtained from the central difference form (3.166), resulting in

$$\frac{d^n}{dt^n}\beta^N(t) = \frac{1}{(N-n)!}\delta^{N+1}t_+^{N-n} \quad \text{for } n = 0, \ldots, N, \tag{5.62}$$

where δ^k denotes the central difference operator (3.167). Thus, the impulse response of the assumed response $H_c^{(N)}(j\omega)$ is represented by

$$h_c^{(N)}(t) = \frac{1}{N!}\delta^{N+1}t_+^N + \sum_{i=1}^{\lfloor\frac{N}{2}\rfloor}(-1)^i c_i \frac{1}{(N-2i)!}\delta^{N+1}t_+^{N-2i}. \tag{5.63}$$

In the following, it is shown that the formulas (5.60) and (5.63) hold for Lagrange interpolators of arbitrary order, and an explicit expression for the constants c_i is derived.

Because the relationships between time- and frequency-domain characteristics are well-established in the theory of spline functions, it is beneficial to express the continuous impulse response of Lagrange interpolation in a form similar to spline basis functions. In [Sch46a], the impulse response of a Lagrange interpolator is represented in form of a central difference operator

$$h_c^{(N)}(t) = \frac{1}{N!} \delta^{N+1} x_+^{[N]},\tag{5.64}$$

exhibiting the same structure as a spline basis function in central difference form (3.166). Here, $x^{[n]}$ denotes the one-sided or truncated central factorial

$$x_+^{[n]} = \begin{cases} x^{[n]}, & x \geq 0 \\ 0, & x < 0 \end{cases}\tag{5.65a}$$

$$= H(x)x^{[n]},\tag{5.65b}$$

where $H(x)$ denotes the Heaviside unit step function (3.170). The central factorial $x^{[n]}$ is defined as [Sch46a]

$$x^{[n]} = \underbrace{\left(x - \tfrac{N-1}{2}\right)\left(x - \tfrac{N-3}{2}\right)\cdots\left(x + \tfrac{N-3}{2}\right)\left(x + \tfrac{N-1}{2}\right)}_{n}\tag{5.66a}$$

$$= \prod_{i=1}^{N}\left(x - \tfrac{N+1}{2} + i\right)\tag{5.66b}$$

$$= \begin{cases} \prod_{i=1}^{\frac{N}{2}}\left(x^2 - \dfrac{(2i-1)^2}{4}\right), & N \text{ even} \\[2mm] x\prod_{i=1}^{\frac{N-1}{2}}\left(x^2 - i^2\right), & N \text{ odd} \end{cases}\tag{5.66c}$$

It is important to note that the above definition differs from other definitions found in the mathematical literature, e.g. [BSSV89], for even orders N.

The formula (5.64) is introduced in [Sch46a] in an informal manner without proof or further references. For this reason, a derivation has been provided in the appendix of [FB09a].

To relate (5.64) to the central difference form of a spline basis function (3.166), it is transformed into a representation based on one-sided power functions (3.169). In a first step, the one-sided central factorial is expressed as a one-sided falling factorial

$$x_+^{[n]} = \begin{cases} \left(x + \tfrac{N-1}{2}\right)^{\underline{N}}, & x \geq 0 \\ 0, & x < 0 \end{cases}\tag{5.67}$$

using the falling factorial function $x^{\underline{n}}$ [GKP06, AS65]

$$x^{\underline{n}} = \underbrace{x(x-1)\cdots(x-n+1)}_{n} = \prod_{l=0}^{n-1}(x-l).\tag{5.68}$$

Because the falling factorials form the generating function for the signed Stirling numbers of the first kind $S_N^{(m)}$ [GKP06, AS65]

$$x^{\underline{N}} = \sum_{m=0}^{N} S_N^{(m)} x^m,$$ (5.69)

the one-sided central factorial (5.65) can be written as

$$x^{[N]} = \begin{cases} \displaystyle\sum_{m=0}^{N} S_N^{(m)} \left(x + \tfrac{N-1}{2}\right)^m, & x \geq 0 \\ 0, & x < 0 \end{cases}$$ (5.70a)

$$= \sum_{m=0}^{N} S_N^{(m)} \left(x + \tfrac{N-1}{2}\right)_+^m.$$ (5.70b)

Utilizing the binomial theorem and reordering the summations with respect to powers of x yields

$$x_+^{[N]} = \sum_{m=0}^{N} S_N^{(m)} \sum_{i=0}^{m} \left(\binom{m}{i} \left(\frac{N-1}{2}\right)^i x_+^{m-i} \right)$$ (5.71a)

$$= \sum_{n=0}^{N} x_+^{N-n} \sum_{m=N-n}^{N} S_N^{(m)} \binom{m}{N-n} \left(\frac{N-1}{2}\right)^{n-N+m}.$$ (5.71b)

Interchanging the order of the summations and substituting the index variable m by $k = N - m$ enables $x^{[N]}$ to be expressed as a polynomial of a one-sided power function with coefficients a_n

$$x_+^{[N]} = \sum_{n=0}^{N} \left(\sum_{k=0}^{n} S_N^{(N-k)} \binom{N-k}{N-n} \left(\frac{N-1}{2}\right)^{n-k} \right) x_+^{N-n}$$ (5.72a)

$$= \sum_{n=0}^{N} a_n x_+^{N-n} \quad \text{with}$$ (5.72b)

$$a_n = \sum_{k=0}^{n} S_N^{(N-k)} \binom{N-k}{N-n} \left(\frac{N-1}{2}\right)^{n-k}.$$ (5.72c)

In this way, the central difference form of the impulse response of Lagrange interpolators (5.64) can be expressed using one-sided power functions, utilizing the linearity of the central difference operator [Nør24]

$$h_c^{(N)}(t) = \frac{1}{N!} \delta^{N+1} \sum_{n=0}^{N} a_n t_+^{N-n}$$ (5.73a)

$$= \frac{1}{N!} \sum_{n=0}^{N} a_n \delta^{N+1} t_+^{N-n}.$$ (5.73b)

The constants a_n fulfill several properties. First, since $S_N^{(N)} = 1$ [AS65, GKP06], it follows that

$$a_0 = S_N^{(N)} \binom{N}{N} \left(\frac{N-1}{2}\right)^0 = 1. \tag{5.74a}$$

Second, the central factorial $t^{[N]}$ consists of even powers of t only if N is even, and of odd powers only if N is odd. This follows immediately from the definition (5.66c). Inspection of (5.73) shows that

$$a_n = 0 \quad \text{for } n \text{ odd}. \tag{5.74b}$$

Comparing (5.73), combined with the stated properties of the constants a_n, to the conjectured form of the continuous impulse response $h_c^{(N)}(t)$ (5.63) confirms the proposed structure of the continuous frequency (5.60)

$$H_c^{(N)}(\mathrm{j}\omega) = \mathrm{sinc}\left(\frac{\omega}{2}\right)^{N+1}\left(1 + \sum_{i=1}^{\lfloor \frac{N}{2} \rfloor} c_i \omega^{2i}\right)$$

and provides an explicit expression for the constants c_i

$$
\begin{aligned}
c_i &= (-1)^i \frac{(N-2i)!}{N!} a_{2i} \\
&= (-1)^i \frac{(N-2i)!}{N!} \sum_{k=0}^{2i} S_N^{N-k} \binom{N-k}{N-2i}\left(\frac{N-1}{2}\right)^{2i-k} \quad \text{for } i = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor.
\end{aligned}
\tag{5.75}
$$

5.3.4 Alternative Derivations

The above proof provides a self-sufficient derivation of $H_c^{(N)}(\mathrm{j}\omega)$ for arbitrary orders and gives explicit expressions for the constants c_i. However, several other approaches to represent this frequency response have been proposed, which will be summarized in the following.

As noted above, the general structure of $H_c^{(N)}(\mathrm{j}\omega)$ (5.60) was known to I.J. Schoenberg in his papers introducing spline interpolation. In [Sch46a], explicit formulas for $H_c^{(N)}(\mathrm{j}\omega)$ are given for the orders $N = 0, \ldots, 3$. In [Sch46b, Lemma 6], it is stated that the basis function $\Gamma_{k,\mu}(x)$, corresponding to the continuous impulse response $h_c(t)$,

$$\Gamma_{k,\mu}(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \left(\frac{2\sin(u/2)}{u}\right)\left\{1 + \gamma_2^{(k)} u^2 + \cdots + \gamma_{2\mu-2}^{(k)} u^{2\mu-2}\right\} e^{\mathrm{j}ux} \, \mathrm{d}u \tag{5.76}$$

is identical to the k-point central interpolation formula, that is, Lagrange interpolation of order $N = k - 1$ with a basepoint set centered around $x = 0$. Here, the index limit μ is given by $\mu = \left\lfloor \frac{k-1}{2} \right\rfloor = \left\lfloor \frac{N}{2} \right\rfloor$. As the integral in (5.76) represents an inverse Fourier transform in the

frequency variable u, its structure is indeed identical to (5.60). Moreover, it is proved that the interpolation condition (3.67) is equivalent to the frequency domain condition

$$\sum_{k=-\infty}^{\infty} H_c\left(j[\omega + 2\pi k]\right) \equiv 1 \quad \text{for } \omega \in \mathbb{R}. \tag{5.77}$$

Inserting the general structure of $H_c^{(N)}(j\omega)$ (5.60) into the interpolation condition (5.77) yields an expression for the coefficients c_i [Sch46b, BTU01]

$$\sum_{i=0}^{\infty} c_i \omega^i = \left(\frac{w/2}{\sin(\omega/2)}\right)^{N+1}, \tag{5.78}$$

that is, the constants c_i are determined by the Maclaurin series expansion of the function

$$f(\omega) = \left(\frac{w/2}{\sin(\omega/2)}\right)^{N+1}. \tag{5.79}$$

The series coefficients can be generated from the series expansion $(x/\sin(x))^k$, which are tabulated in [Nør24] for low orders k.

An approach to relate time- and frequency-domain characteristics of interpolation and approximation methods originates from a paper by Strang and Fix [SF73], which is concerned with the analysis of the finite element method. This approach is motivated from approximation theory. For this reason, it is somewhat different from the filtering processes considered here, which focus on calculating the value of a function at arbitrary points from a set of function values. Instead, the Strang and Fix theory considers the determination of a function $u(x)$ as a linear combination from a space of functions

$$u(x) = \sum_k c[k] \phi_k(x) \tag{5.80}$$

such that the approximation error becomes minimal with respect to a given norm $\|\cdot\|_p$

$$\underset{c[k]}{\text{minimize}} \, \|u(x) - f(x)\|_p. \tag{5.81}$$

The basis functions $\phi_k(x)$ generally have a finite support. They are generated by shifting a single function $\phi(x)$ by an integer value k [SF73, BTU01]

$$\phi_k(x) = \phi(x - k). \tag{5.82}$$

In this way, (5.80) can be stated as

$$u(x) = \sum_k c[k] \phi(x - k). \tag{5.83}$$

This representation matches the discrete filtering process (3.96) based on a continuous-time impulse response $h_c(t)$ very closely.

An important concept in approximation theory is the approximation order of a kernel $\phi(x)$. For a space of functions $f(x)$ that meets certain requirements such as smoothness, which are inherently fulfilled by the bandlimited functions considered here, the approximation order L is defined by

$$\|u(x) - f(x)\|_p \le c \times T^L \times \|f(x)\|_p \qquad \text{for } T \to 0. \tag{5.84}$$

Here, T is the sampling period and c is a finite real number. That is, a method is of approximation order L if the error decreases at least proportional to T^L when the sampling period approaches zero.

The approximation order depends only on the function $\phi(x)$. In [SF73], a set of equivalent conditions is stated, which are commonly referred to as the Strang and Fix conditions [Boo90, BTU01, Mei02, BTU03]:

- The method has an approximation order of L, i.e. it fulfills (5.84).
- The Fourier transform of $\phi(x)$, denoted $\Phi(\omega)$, has zeros of order L at all integer multiples of 2π, that is

$$\Phi(0) \ne 0 \tag{5.85a}$$

$$\Phi^{(l)}(2\pi k) = 0 \qquad l = 0, 1, \ldots, L-1 \text{ and } k \in \mathbb{Z}, \tag{5.85b}$$

  where $\Phi^{(l)}(\omega)$ denotes the l-th derivative of $\Phi(\omega)$ with respect to ω.
- Any monomial of the form x^n of orders up to $L-1$ can be reproduced exactly. That is, coefficients $c[n,k]$ exist such that

$$\sum_k c[n,k]\phi(x-k) = x^n \qquad \text{for } n = 0, 1, \ldots, L-1 \tag{5.86}$$

  holds exactly. Because any polynomial of order $L-1$ can be stated as a linear combination of these monomials, this condition implies that a method has approximation order L if and only if it is capable of reproducing all polynomials of order up to $L-1$.
- The first L discrete moments are constants, that is, a set of constants $\mu_n \in \mathbb{R}$ exists such that

$$\sum_k (x-k)^n \phi(x-k) = \mu_n \qquad \text{for } n = 0, 1, \ldots, L-1. \tag{5.87}$$

In signal processing, the application of the Strang and Fix conditions was pioneered by [BU99, BTU01, BTU03]. However, mainly due to their application in image processing, these references have a strong emphasis on the L_2 norm. In [BTU01], it is shown that the class of interpolation kernels of approximation order L with respect to the L_2 norm is a subset of the vector space formed by the time-domain derivatives of the spline basis functions of order $L-1$

$$\phi(t) = \sum_{n=0}^{L-1} \gamma_n \frac{\mathrm{d}^n}{\mathrm{d}t^n} \beta^{L-1}(t-a). \tag{5.88}$$

For the resampling functions considered here, the displacement a is zero due to the symmetry of the continuous impulse response $h_c(t)$. For the same reason, the coefficients γ_n are nonzero
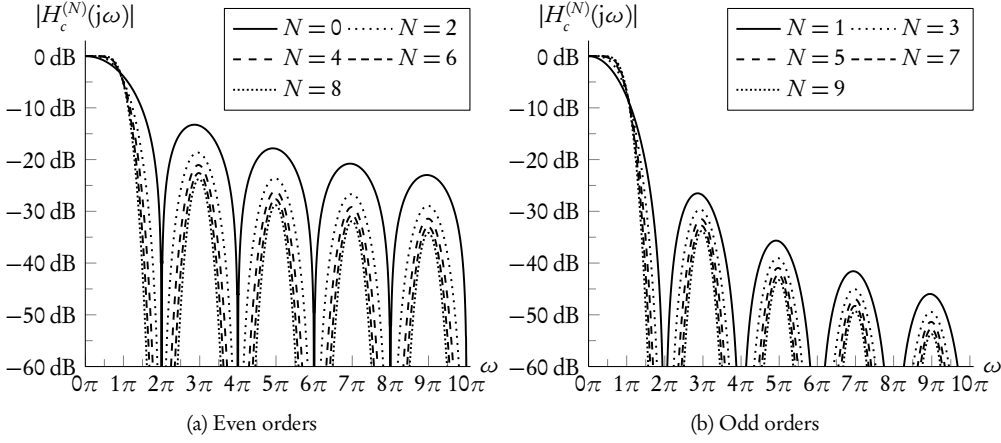
Figure 5.5: Continuous frequency response of Lagrange interpolation filters for orders $N = 0, 1, \ldots, 9$. The graphs are grouped into even and odd orders N due to their qualitative differences.

only for even n. The most obvious member of this class is the spline basis function of order $L - 1$, corresponding to the parameter set $\gamma_0 = 1$, $\gamma_n = 0$ for $n = 1, \ldots, L - 1$.

By definition, N-th order Lagrange interpolation reproduces a polynomial of order N exactly. It is therefore a member of the vector space spanned by (5.88). Hence, this representation poses an alternative way to confirm the general structure of the frequency response (5.60). In [BTU01], the coefficients γ_n are determined by imposing the frequency-domain interpolation condition (5.77) on (5.88) similar to [Sch46b], consequently resulting in an equivalent expression.

Several remarkable conclusions follow from (5.88). First, all resampling functions with support L and approximation order L with respect to the L_2 norm are piecewise polynomial functions. Thus, the widespread use of piecewise polynomials in interpolation and SRC is justified not only by their computational efficiency, but also by the approximation power of this class of functions. Second, the minimum support of any resampling functions of approximation order L is L, and all functions exhibiting this property are described by (5.88). For this reason, this class of functions has been termed *maximal-order interpolation of minimal support (MOMS)* in [BTU01].

5.3.5 Discussion

Besides its utility for the stable and efficient evaluation of $H_c^{(N)}(j\omega)$, the proposed closed-form expression enables an exact characterization of several properties of Lagrange interpolation.

In the literature on sample rate conversion, the zero locations of $H_c^{(N)}(j\omega)$ are typically described only in a qualitative way using statements as "the zeroes of the system function tend to be clustered about those frequencies [integer multiples of $2\pi/T_i$]" [SR73, HF00, Hen02]. However, with the proposed formula, it becomes apparent that $H_c^{(N)}(j\omega)$ has zeros of order $N + 1$ exactly at the locations $\omega = 2\pi k$, $k = \pm 1, \pm 2, \ldots$. This result is also evident from the Strang and Fix conditions introduced above. Moreover, as the coefficients c_i are strictly positive

Figure 5.6: Hybrid analog/digital model of an ASRC system based on integer-ratio oversampling and Lagrange interpolation.

and the polynomial of (5.60) contains only even powers, $H_c(j\omega)$ has no additional real zeros.

The limited image attenuation of Lagrange interpolators is also described only qualitatively in most references, e.g. [Hen02]. Inspection of (5.60) shows that the attenuation towards higher frequencies is of order $O(\omega^{-1})$ or $O(\omega^{-2})$ for N even or odd, respectively. Thus, the asymptotic image attenuation does not increase systematically as the order of interpolation rises. In addition, this asymptotic attenuation confirms the qualitative differences between even- and odd-order interpolation, which have been stated by numerous authors, e.g. [Väl95a, SR73, EGH93], analytically.

The continuous frequency responses for some low interpolation orders are shown in Figure 5.5, also illustrating the qualitative differences due to the parity of N. While the stopband attenuation is inferior for N even, the passband response is significantly more flat in this case. This behavior is similar to that of the amplitude response of fractional delay filters based on Lagrange interpolation [Väl95a, Smi10a].

5.4 Overall Optimization of Oversampling+Lagrange Structures

As characterized in Section 3.6.5, structures based on integer-ratio sample rate conversion and a continuous-time resampling filter operating on the oversampled input are widely used in ASRC. Nonetheless, design for these structures is basically confined to the design of conventional integer-ratio SRC and the use of fixed continuous-time resampling filters as described in Section 3.8.3. As stated there, Lagrange interpolators are a very common choice for the continuous-time resampling filter in this combination. The resulting structure is denoted *Oversampling+Lagrange* in the following.

The resulting structure is a cascade of two components that are typically designed independently, thus prohibiting a systematic design procedure based on a set of design objectives and constraints. Moreover, the obtained designs are generally not optimal with respect to a prescribed error norm such as the weighted least squares or the Chebyshev norm.

To overcome these deficiencies, an overall optimization scheme for this class of resampling filters is presented. This method has been initially proposed in [FB09b]. Starting from a closed-form frequency domain description which is based on the continuous frequency response of Lagrange interpolation derived in the preceding section, the design objective is formulated and solved as a convex optimization problem.

5.4.1 Closed-Form Description

The general structure of the system has been described in Section 3.6.5. Its representation in the hybrid analog/digital model (e.g. [Eva00b, Eva03]) is shown in Figure 5.6. The continuous
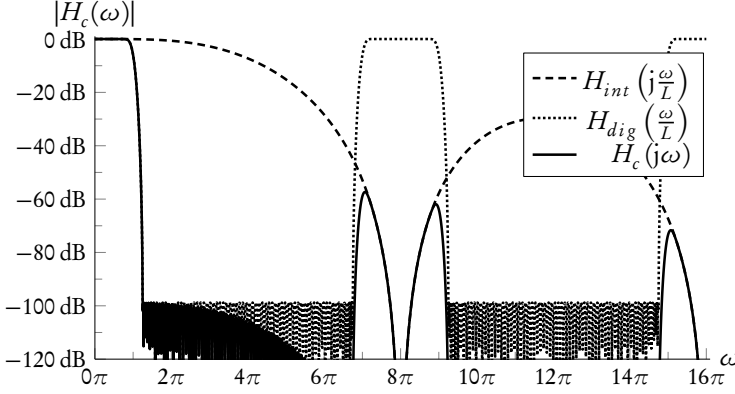
Figure 5.7: Continuous frequency response of a ASRC system based on oversampling and La-
grange interpolation. Oversampling ratio $L = 4$, Lagrange interpolation order
$N_{int} = 3$, discrete-time filter order $N_{dig} = 95$ designed according to L_∞ norm, cutoff
frequency $\omega_c = 0.75\pi$.

frequency response of this system is represented by

$$H_c(\mathrm{j}\omega) = \frac{1}{L} H_{dig}\left(\mathrm{e}^{\frac{\mathrm{j}\omega}{L}}\right) H_{int}\left(\frac{\mathrm{j}\omega}{L}\right), \tag{5.89}$$

where $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$ is the discrete-time frequency response of the anti-imaging filter of the integer
SRC system and $H_{int}(\mathrm{j}\omega)$ denotes the continuous-time frequency response of the Lagrange
interpolator. As in the preceding sections, the input sampling frequency is normalized to $T_i = 1$,
thus no magnitude scaling is necessary in (5.89). Likewise, ω denotes the angular frequency
normalized to $T_i = 1$. In this way, the continuous frequency response is formed by the cascade
of the filters $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$ and $H_{int}(\mathrm{j}\omega)$, while the oversampling ratio L results in a scaling of the
frequency variable. The magnitude scaling of $\frac{1}{L}$ originates from the frequency response of the
sample rate expander as described in Section 3.4.2.

The continuous frequency of a typical ASRC system based on this structure is shown in
Figure 5.7. Here, the contributions of the cascaded filters to the system response $H_c(\mathrm{j}\omega)$ become
apparent. The discrete-time and thus periodic anti-imaging filter $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$ attenuates the images
in the intervals $[(2kL+1)\pi, (2[k+1]L-1)\pi]$, $k = 0, \pm 1, \pm 2, \ldots$. In contrast, the non-periodic
frequency response $H_{int}(\mathrm{j}\omega)$ of the Lagrange interpolator dampens the spectral replications of
the passband of $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$, which are centered around integer multiples of $2\pi L$.

In most applications of rational SRC [SR73, CR83] as well as in ASRC (e.g. [Eva00b]), linear-
phase FIR filters are strongly preferred for the discrete-time anti-imaging filter $H_{dig}(\mathrm{e}^{\mathrm{j}\omega})$. The
main reasons are its exact linear phase and the existence of efficient polyphase realizations.
Therefore, linear-phase FIR filters are considered here exclusively.

A convenient representation for the frequency response of linear-phase filters has been intro-
duced in Section 3.6.2, which partitions the frequency response into a pure delay term and a

non-causal frequency response $H(\omega)$.

$$H_{dig}(e^{j\omega}) = e^{-j\frac{N}{2}\omega} H_{dig}(\omega) \qquad \text{with} \tag{5.90a}$$

$$H_{dig}(\omega) = \sum_{n=0}^{N'_{dig}} b[n] \operatorname{trig}(n,\omega) \qquad \text{where } N'_{dig} = \left\lfloor \frac{N}{2} \right\rfloor . \tag{5.90b}$$

(3.121). As argued in Section 3.5.5, the delay term is typically compensated by an implementation delay to ensure causality. So, only the non-causal frequency response $H_{dig}(\omega)$ is considered in the following. For use in integer-ratio SRC, only even-symmetric linear-phase FIR filters, that is, of type I or II, are of interest. Thus, $H_{dig}(\omega)$ also represents the zero-phase frequency response of $H_{dig}(e^{j\omega})$, and the basis function $\operatorname{trig}(n,\omega)$ simplifies to

$$\operatorname{trig}(n,\omega) = \begin{cases} 1, & N_{dig} \text{ even}, n = 0 \\ 2\cos(n\omega), & N_{dig} \text{ even}, n > 0 , \\ 2\cos\left(\left[n + \frac{1}{2}\right]\omega\right), & N_{dig} \text{ odd} \end{cases} \tag{5.90c}$$

where N_{dig} denotes the order of the discrete-time prefilter $H_{dig}(e^{j\omega})$.

A closed-form expression for the continuous-time frequency response of Lagrange interpolation has been introduced in Section 5.3. Applying this description and (5.90) to (5.89) yields a closed-form expression for the continuous-time frequency response of the structure based on oversampling and Lagrange interpolation.

$$H_c(j\omega) = \sum_{n=0}^{N'_{dig}} b[n] G_o(n,\omega) \quad \text{with} \tag{5.91a}$$

$$G_o(n,\omega) = \frac{1}{L} \operatorname{trig}\left(n, \frac{\omega}{L}\right) \left(1 + \sum_{i=1}^{\left\lfloor \frac{N_{int}}{2} \right\rfloor} c_i \left(\frac{\omega}{L}\right)^{2i}\right) \operatorname{sinc}\left(\frac{\omega}{2L}\right)^{N_{int}+1}, \tag{5.91b}$$

where the functions $G_o(n,\omega)$ form the basis functions of the continuous frequency response. N_{int} denotes the order of the continuous-time resampling function $H_{int}(j\omega)$, that is, the order of the Lagrange interpolator.

5.4.2 Optimization problem

The frequency response $H_c(j\omega)$ is linear with respect to the coefficients $b[n]$. Consequently, numerous efficient design methods are applicable. Within this work, we focus on methods based on convex optimization as motivated in Section 5.2.4. As in case of the modified Farrow structure, the design is performed on a discretized version of $H_c(j\omega)$. So, the considerations concerning the discretization grid apply here, too. However, due to the periodicity of $H_{dig}(e^{j\omega})$ and the attenuation of $H_{int}(j\omega)$ towards higher frequencies, it can be shown that an upper bound $\omega_{up} = 2\pi L$ for the approximation region is sufficient. For optimization with respect to the L_∞ norm, this identity holds exactly. In contrast, as argued in Section 5.2.4, a numerical approximation of

the L_2 error depends on the upper limit of the discretization grid. Nonetheless, the differences between designs based on the minimum admissible frequency limit and other admissible values are negligible because of the attenuation caused by $H_{int}(j\omega)$ for high frequencies.

Using a discrete grid of frequencies $\mathbf{X}$ (5.35) approximating either a uniform or a nonuniform stopband specification, the discretized frequency response is given by

$$\mathbf{H}_c = \mathbf{G}_o \mathbf{b} \qquad \qquad \text{with} \qquad (5.92\text{a})$$

$$\mathbf{G}_o = \begin{bmatrix} G_o(0,\omega_0) & G_o(1,\omega_0) & \cdots & G_o(N',\omega_0) \\ G_o(0,\omega_1) & G_o(1,\omega_1) & \cdots & G_o(N',\omega_1) \\ \vdots & \vdots & \ddots & \vdots \\ G_o(0,\omega_K) & G_o(1,\omega_K) & \cdots & G_o(N',\omega_K) \end{bmatrix} \quad \text{and} \qquad (5.92\text{b})$$

$$\mathbf{b} = \begin{bmatrix} b[0] & b[1] & \cdots & b[N'_{dig}] \end{bmatrix} . \qquad (5.92\text{c})$$

Optimization with respect to the L_∞ norm is formulated in a CVX program

```
cvx_begin
    variable  b(N'_dig +1);
    minimize ( norm ( W . (G_o b–Ĥ_c), Inf ) );
cvx_end
```

where $\mathbf{W}$ is a vector of weights (5.42) and $\widehat{\mathbf{H}}_c$ (5.41) denotes the discretized desired frequency response. As in case of the modified Farrow structure, optimization with respect to a weighted least squares criterion is achieved by replacing the norm function with $\mathbf{norm}(\cdot, 2)$. Other design objectives, such as minimizing the stopband error norm for a given maximum passband error δ_p, or minimizing the passband error for a fixed stopband error limit δ_s, are applied as described in Section 5.2.4.

In contrast, the specification of conditions related to time-domain properties of the continuous-time impulse response $h_c(t)$ is different from design methods for the modified Farrow structure. The properties of continuity and regularity are determined by the Lagrange interpolator and cannot be controlled by design specifications for the discrete-time prefilter. Following the characterization of Lagrange interpolators in Section 5.3.5, the impulse response $h_c(t)$ of the complete system is continuous and of regularity order 0 for N odd, That is, it has no continuous derivative. For N even, $h_c(t)$ is not continuous.

Because Lagrange interpolation inherently fulfills the interpolation condition, the adherence to this condition depends on the ability of the discrete-time filter $H_{dig}(e^{j\omega})$ to reproduce input samples exactly. Formally, this requires the zeroth polyphase branch of $H_{dig}(e^{j\omega})$ to form a discrete-time Dirac impulse sequence [CR83]. For the linear-phase FIR filters considered here, this requirement cannot be fulfilled exactly for type II filters, that is, N_{dig} odd. In contrast, the interpolation condition can be trivially enforced for discrete-time anti-imaging filters of even order N_{dig} by restricting the coefficients $\mathbf{b}[n]$ such that the zeroth polyphase branch forms a symmetric unit impulse sequence.
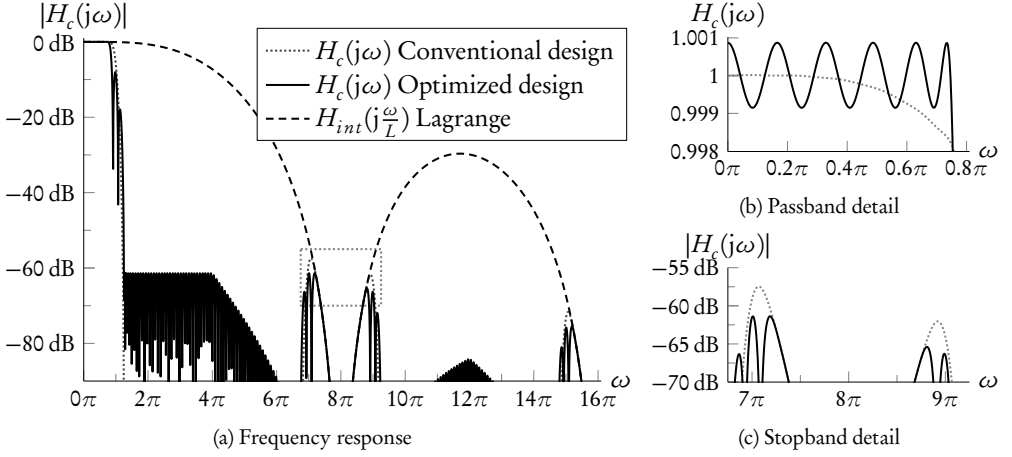
Figure 5.8: Design example for proposed overall optimization scheme for Oversampling+Lagrange structures, L_∞ design. Parameters $L = 4$, $N_{dig} = 95$, $N_{int} = 3$, $\omega_c = 0.75\pi$, $W(X) = 1$. Comparison to conventional design.

5.4.3 Results

In Figure 5.8, the continuous frequency response of a structure designed with the proposed overall optimization scheme is compared to a conventional design with an equiripple lowpass design for $H_{dig}(e^{j\omega})$. It is observed that the maximum passband error is decreased to about 47.7 %, while the minimum stopband attenuation improves by about 4.03 dB.

This example also illustrates the causes for this performance improvement. In the stopband, the performance gain basically results from an increased attenuation of $H_c(j\omega)$ in the intervals $(2\pi[nL-1]+\omega_c, 2\pi nL-\omega_c) \cup (2\pi nL+\omega_c, 2\pi[nL+1]-\omega_c)$, $n = \pm 1, \pm 2, \dots$. These regions correspond to the images of the transition band of $H_{dig}(e^{j\omega})$. In the passband, the frequency response of the conventional design shows a damping towards higher frequencies that is characteristic to Lagrange interpolation. In contrast, the proposed design method is capable of compensating this error and achieves an optimal passband performance with respect to the given design specification. In the considered example, the constant weighting function results in equal maximum passband and stopband errors.

Table 5.3 shows Oversampling+Lagrange designs with variations in the parameters L, N_{dig}, N_{int} for $\omega_c = 0.75\pi$. It is observed that the proposed method increases the minimum stopband attenuation A_s in virtually all cases. The stopband attenuation is a monotonic function of the prefilter order N_{dig} for fixed values of L and N_{int}, but the magnitude of the improvement over the conventional design depends on the parameters used. If the filter order N_{dig} is relatively small for a given oversampling ratio L and Lagrange interpolation order N_{int}, the achievable performance is dominated by the limited image rejection of the discrete-time anti-imaging filter, and thus the improvement is negligible. On the other hand, if the order of the prefilter is very large, the achievable performance is restricted by the limited image attenuation of the Lagrange interpolator in the frequency intervals corresponding to the passband images of $H_{dig}(e^{j\omega})$. This

Parameters			Conventional design		Proposed design		Improvement		
L	N_{int}	N_{dig}	$\delta_p^{(conv)}$	$A_s^{(conv)}$	$\delta_p^{(opt)}$	$A_s^{(opt)}$	$\dfrac{\delta_p^{(opt)}}{\delta_p^{(conv)}}$	$A_s^{(opt)} - A_s^{(conv)}$	
4	3	63	$2.12 \cdot 10^{-3}$	57.86 dB	$1.05 \cdot 10^{-3}$	59.54 dB	49.65 %	1.68 dB	
4	3	95	$1.79 \cdot 10^{-3}$	57.34 dB	$8.53 \cdot 10^{-4}$	61.38 dB	47.72 %	4.03 dB	
4	3	127	$1.78 \cdot 10^{-3}$	56.97 dB	$7.85 \cdot 10^{-4}$	62.10 dB	44.16 %	5.13 dB	
4	3	128	$1.78 \cdot 10^{-3}$	56.97 dB	$7.83 \cdot 10^{-4}$	62.13 dB	44.05 %	5.16 dB	
4	3	191	$1.78 \cdot 10^{-3}$	56.53 dB	$7.33 \cdot 10^{-4}$	62.69 dB	41.28 %	6.16 dB	
4	3	255	$1.78 \cdot 10^{-3}$	60.02 dB	$7.16 \cdot 10^{-4}$	62.90 dB	40.30 %	2.88 dB	
4	4	255	$7.54 \cdot 10^{-5}$	64.95 dB	$3.81 \cdot 10^{-4}$	68.37 dB	505.90 %	3.43 dB	
4	5	255	$1.24 \cdot 10^{-4}$	80.98 dB	$5.86 \cdot 10^{-5}$	84.65 dB	47.16 %	3.67 dB	
4	6	255	$5.66 \cdot 10^{-6}$	83.80 dB	$3.24 \cdot 10^{-5}$	89.80 dB	571.83 %	6.00 dB	
4	7	255	$9.16 \cdot 10^{-6}$	96.93 dB	$5.52 \cdot 10^{-6}$	105.16 dB	60.29 %	8.23 dB	
2	5	255	$6.63 \cdot 10^{-3}$	47.94 dB	$3.22 \cdot 10^{-3}$	49.84 dB	48.63 %	1.90 dB	
3	5	255	$6.66 \cdot 10^{-4}$	59.71 dB	$3.02 \cdot 10^{-4}$	70.40 dB	45.37 %	10.69 dB	
4	5	255	$1.24 \cdot 10^{-4}$	80.98 dB	$5.86 \cdot 10^{-5}$	84.65 dB	47.16 %	3.67 dB	
6	5	255	$1.13 \cdot 10^{-5}$	97.01 dB	$6.74 \cdot 10^{-6}$	103.44 dB	59.72 %	6.43 dB	

Table 5.3: Design experiments for Oversampling+Lagrange structures, L_∞ design. Comparison between conventional design and the proposed overall optimization scheme. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$.

implies a practical upper bound on the achievable quality for a given combination of L and N_{int} that cannot be surpassed by a further increase of N_{dig}. In contrast, for the conventional design, the stopband error does not decrease monotonically as the filter order N_{dig} increases.

In most design examples, also the passband error decreases compared to the conventional design. However, for N_{int} even, that is, in the examples with $N_{int} = 4$ or $N_{int} = 6$ for $L = 4$ and $N_{dig} = 255$, δ_p increases, which is indicated by a ratio $\delta_p^{(opt)}/\delta_p^{(conv)}$ exceeding 100 %. However, the good passband quality of the conventional design originates in the superior passband response of even-order Lagrange interpolation (see Section 5.3.5). In contrast, the increased error for the proposed design method is caused by the design objective $W(\omega) = 1$, which enforces an equal weighting between passband and stopband error. In fact, this equality between δ_p and δ_s holds for all design examples. This property again demonstrates the ability of the proposed method to yield designs that are optimal with respect to an arbitrary set of design specifications. With the conventional approach, such a purposeful design is not possible.

The design example of Figure 5.8 is repeated with respect to the L_2 norm with uniform weighting. The resulting continuous frequency response is depicted in Figure 5.9. The discrete-time prefilter for the conventional design used for comparison has been obtained using a least-squares lowpass design with uniform error weighting.

Compared to the conventional design, the L_2 norm is decreased to 20.4 % for the passband error $L_2(X_p)$ and to 69.9 % for the stopband error $L_2(X_s)$. Moreover, it is observed that both the maximum passband and stopband errors are decreased compared to the conventional L_2 design.

Table 5.4 shows a set of design experiment for L_2 design, using the parameter variations of Table 5.3. The upper limit ω_{up} for the approximation region X is set to $2\pi L$ both for the design
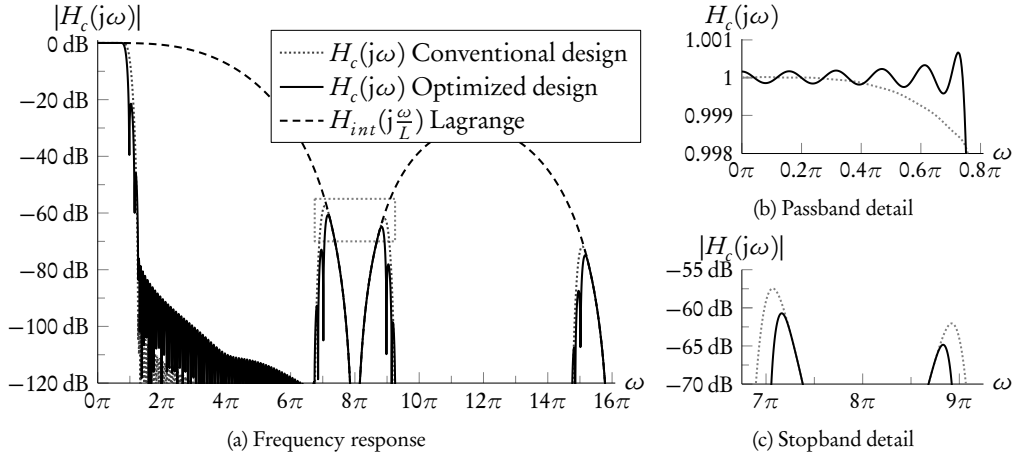
Figure 5.9: Design example for proposed overall optimization scheme for Oversampling+Lagrange structures, L_2 design. Parameters $L = 4$, $N_{dig} = 95$, $N_{int} = 3$, $\omega_c = 0.75\pi$, $W(X) = 1$, upper frequency limit $\omega_{up} = 2\pi L$. Comparison to conventional design.

and for the calculation of the stopband error norm $L_2(X_s)$.

As in case of the minimax design specification, the L_2 stopband error generally decreases compared to the conventional design method. Likewise, the passband error is decreased considerably for N_{int} odd, while it remains approximately constant or increases for even orders of Lagrange interpolation. As in case of the L_∞ norm, this behavior is due to the uniform error weighting in use.

5.4.4 Summary

A design method that performs an overall optimization of Oversampling+Lagrange structures with respect to selectable error norms has been proposed. While an improvement compared to conventionally designed structures is achieved for virtually all design specification, the main advantage of this method is the ability to perform a purposeful design that is optimal with respect to a given design specification and optionally adheres to a set of design constraints.

5.5 Optimization of ASRC Structures Based on Oversampling and Fixed Resampling Functions

The preceding section has shown that the performance of ASRC algorithms based on oversampling and Lagrange interpolation can be improved by design methods for the discrete-time prefilter $H_{dig}(e^{j\omega})$ that take the overall frequency response of the structure into account. However, it also revealed shortcomings of the Oversampling+Lagrange structure. In particular, the limited attenuation of the continuous-time resampling filter $H_{int}(j\omega)$ in the passband image regions of $H_{dig}(e^{j\omega})$ constitutes an upper bound for the achievable stopband attenuation.

Parameters			Conventional design		Proposed design		Improvement	
L	N_{int}	N_{dig}	$L_2^{(conv)}(X_p)$	$L_2^{(conv)}(X_s)$	$L_2^{(opt)}(X_p)$	$L_2^{(opt)}(X_s)$	$\frac{L_2^{(opt)}(X_p)}{L_2^{(conv)}(X_p)}$	$\frac{L_2^{(opt)}(X_s)}{L_2^{(conv)}(X_s)}$
4	3	63	$9.64 \cdot 10^{-4}$	$1.15 \cdot 10^{-3}$	$4.13 \cdot 10^{-4}$	$1.13 \cdot 10^{-3}$	42.83 %	98.06 %
4	3	95	$9.15 \cdot 10^{-4}$	$1.15 \cdot 10^{-3}$	$1.86 \cdot 10^{-4}$	$8.05 \cdot 10^{-4}$	20.36 %	69.88 %
4	3	127	$9.15 \cdot 10^{-4}$	$1.17 \cdot 10^{-3}$	$9.13 \cdot 10^{-5}$	$6.56 \cdot 10^{-4}$	9.98 %	56.26 %
4	3	128	$9.15 \cdot 10^{-4}$	$1.16 \cdot 10^{-3}$	$8.35 \cdot 10^{-5}$	$6.50 \cdot 10^{-4}$	9.12 %	56.12 %
4	3	191	$9.15 \cdot 10^{-4}$	$9.46 \cdot 10^{-4}$	$6.80 \cdot 10^{-5}$	$5.27 \cdot 10^{-4}$	7.43 %	55.69 %
4	3	255	$9.15 \cdot 10^{-4}$	$7.19 \cdot 10^{-4}$	$5.80 \cdot 10^{-5}$	$4.76 \cdot 10^{-4}$	6.34 %	66.15 %
4	4	255	$3.23 \cdot 10^{-5}$	$3.96 \cdot 10^{-4}$	$3.08 \cdot 10^{-5}$	$2.47 \cdot 10^{-4}$	95.32 %	62.38 %
4	5	255	$5.33 \cdot 10^{-5}$	$6.13 \cdot 10^{-5}$	$4.84 \cdot 10^{-6}$	$3.79 \cdot 10^{-5}$	9.09 %	61.84 %
4	6	255	$2.12 \cdot 10^{-6}$	$3.54 \cdot 10^{-5}$	$3.38 \cdot 10^{-6}$	$2.07 \cdot 10^{-5}$	159.80 %	58.54 %
4	7	255	$3.43 \cdot 10^{-6}$	$6.00 \cdot 10^{-6}$	$5.00 \cdot 10^{-7}$	$3.64 \cdot 10^{-6}$	14.56 %	60.58 %
2	5	255	$2.92 \cdot 10^{-3}$	$2.98 \cdot 10^{-3}$	$1.38 \cdot 10^{-4}$	$1.64 \cdot 10^{-3}$	4.72 %	55.21 %
3	5	255	$2.88 \cdot 10^{-4}$	$3.09 \cdot 10^{-4}$	$1.54 \cdot 10^{-5}$	$1.75 \cdot 10^{-4}$	5.36 %	56.53 %
4	5	255	$5.33 \cdot 10^{-5}$	$6.13 \cdot 10^{-5}$	$4.84 \cdot 10^{-6}$	$3.79 \cdot 10^{-5}$	9.09 %	61.84 %
6	5	255	$4.82 \cdot 10^{-6}$	$1.03 \cdot 10^{-5}$	$9.71 \cdot 10^{-7}$	$5.45 \cdot 10^{-6}$	20.15 %	53.11 %

Table 5.4: Design experiments for Oversampling+Lagrange structures, L_2 design. Comparison between conventional design and the proposed overall optimization scheme. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$, $\omega_{up} = 2\pi L$.

This limited stopband attenuation of Lagrange interpolation has been noted by several authors (e.g. [SR73, Hen02]) and has been characterized in Section 5.3.5.

On the other hand, the favorable properties Lagrange interpolation possibly do not yield an advantage in ASRC systems, especially in combination with oversampled signals. First, the strict adherence to the interpolation condition is of minor importance in many signal processing applications. Second, the relatively flat frequency response of Lagrange interpolation in the passband is not required here, since the passband of the resulting system is formed from a fraction of the passband of the Lagrange interpolator only. In this way, the properties of Lagrange interpolation are likely to bind some degrees of freedom in the specification of the continuous resampling function without significant benefit.

On the contrary, many deficiencies of the continuous resampling function $H_{int}(j\omega)$, especially in the passband region, can be rectified by choosing an appropriate prefiltering function. In a way, this reasoning resembles a concept proposed for interpolation in the field of image processing [BTU99, BTU01]: Instead of requiring the interpolation kernel, corresponding to the continuous resampling function $H_{int}(j\omega)$, to interpolate the input sequence $x[n]$ exactly, an interpolation kernel is sought that improves the quality of interpolation for some sequence $c[n]$ that is generated from $x[n]$. The generation of $c[n]$ can be considered as a filtering operation without loss of generality. Although the original formulation of this approach does not involve oversampling, while the adherence to the interpolation condition is of less importance in ASRC applications, the basic idea is identical. Instead of requiring the continuous-time resampling filter $H_{int}(j\omega)$ to yield an optimal performance for an ideally oversampled input sequence, a resampling filter is utilized that improves the overall quality of the system for a specifically matched oversampled sequence generated from this input signal $x[n]$ by discrete-time filtering.

The overall optimization scheme proposed in the preceding section can be used with arbitrary choices for $H_{int}(j\omega)$, enabling the design of the discrete-time prefilter optimally suited to this resampling filter for a selectable design objective.

5.5.1 B-Spline Basis Functions

As shown in Section 5.3, the continuous frequency response Lagrange interpolation has the general structure

$$H_c^{(N)}(j\omega) = \operatorname{sinc}\left(\frac{\omega}{2}\right)^{N+1}\left(1 + \sum_{i=0}^{\left[\frac{N}{2}\right]} c_i\,\omega^{2i}\right)$$

$$= B^N(\omega)\left(1 + \sum_{i=0}^{\left[\frac{N}{2}\right]} c_i\,\omega^{2i}\right).$$

That is, $H_c^{(N)}(j\omega)$ is formed by the Fourier transform of the B-spline basis function of order N (3.178) scaled by a polynomial in ω. As argued above, this polynomial causes the limited image attenuation of order $O(\omega^{-1})$ and $O(\omega^{-2})$ for N even and odd, respectively.

It is therefore an intuitive choice to omit this polynomial and to use a spline basis function $\beta^N(t)$ as continuous-time resampling filter $H_{int}(j\omega)$. In this way, the frequency response $H_{int}(j\omega)$ has an asymptotic image attenuation of $O(\omega^{-(N+1)})$. The resulting structure is referred to as *Oversampling+B-spline* in the following.

The use of B-splines for ASRC in combination with oversampling has been considered by several authors, e.g. [ZB94, WBJ99, Eva00b, Ves99, VS07]. As pointed out in Section 3.6.4, the application of spline basis functions invariably requires a prior filtering operation, either to ensure the adherence to the interpolation condition or to compensate the attenuation towards higher frequencies in approximation applications. In the mentioned references, this requirement is accounted for in varying degrees and handled by different measures. In [WBJ99], the relatively severe passband attenuation is noted and the use of a compensation filter is suggested. Discrete inverse B-splines, resulting in cardinal spline interpolation [UAE91, Uns99], are used in [Eva00b], while the application of causal FIR filters to approximate these discrete inverse B-spline filters is proposed in [ZB94, Ves99, VS07]. However, neither of these proposals appear to consider the effects of oversampling on the requirements of the compensation filter.

In contrast to these approaches, the proposed overall optimization scheme shows three major advantages. First, the prefiltering operation for spline approximation is integrated into the discrete-time filter $H_{dig}(e^{j\omega})$, thus saving an additional processing stage and potentially reducing the computational effort and/or the implementation latency. Second, due to the overall optimization approach, the prefiltering operation is adapted to the design specification, that is, the oversampling ratio L and the cutoff frequency ω_c. Finally, as in case of the proposed design method for Oversampling+Lagrange structures, the resulting system is optimal with respect to a given norm and additional design constraints specified.

The design method is only a slight adaptation to the overall optimization scheme proposed in Section 5.4.2. In the basis function $G_o(n, \omega)$, (5.91b), the term representing the continuous-time
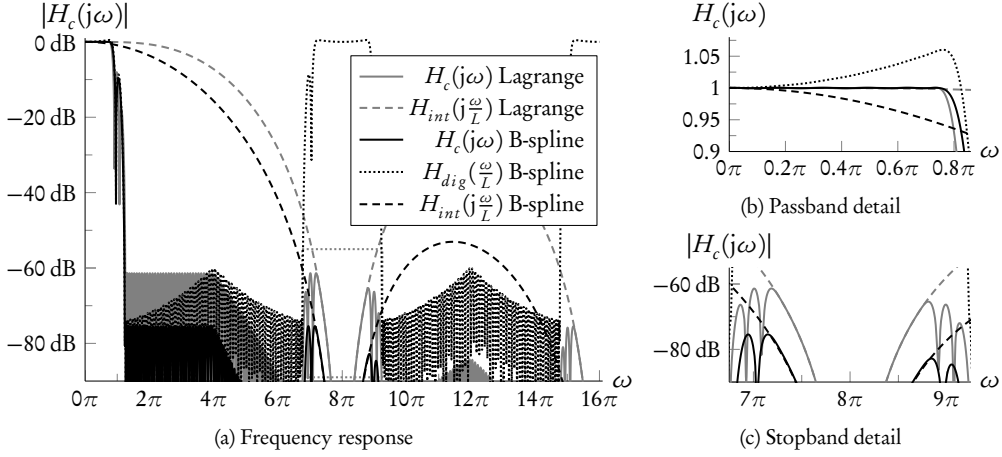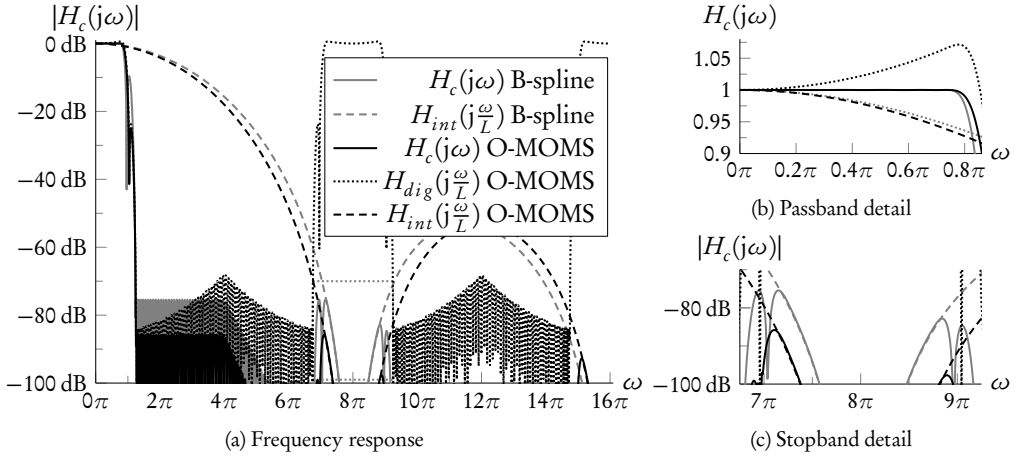
Figure 5.10: Design example for Oversampling+B-spline structure. Parameters $L = 4$, $N_{int} = 3$, $N_{dig} = 95$, $\omega_c = 0.75\pi$. Design with overall optimization scheme, L_∞ norm, $W(X) = 1$. Comparison to Oversampling+Lagrange design.

resampling function $H_{int}(j\omega)$ is replaced by the continuous-time frequency response of the B-spline basis function (3.178), yielding

$$G_o(n, \omega) = \frac{1}{L} \operatorname{trig}\left(n, \frac{\omega}{L}\right) \operatorname{sinc}\left(\frac{\omega}{2L}\right)^{N_{int}+1}. \tag{5.93}$$

The resulting discrete transfer matrix $\mathbf{G}_o$ (5.92b) can be used to formulate the optimization problem, which is solved as described in Section 5.4.

As in case of Oversampling+Lagrange structures, the continuity and regularity of the overall system follows from the respective properties of the continuous-time resampling filter. Thus, the continuous impulse response $h_c(t)$ for Oversampling+B-spline structures is of class C^{N-1}. That is, it is continuous for $N \geq 1$ and has $N - 1$ continuous derivatives.

To impose the interpolation condition, an underdetermined system of linear equations in the coefficients $b[n]$ of $H_{dig}(e^{j\omega})$ must be incorporated into the design as an additional equality constraint. However, identical to Lagrange interpolation, exact compliance to the interpolation condition is possible only for even N_{dig}. For N_{dig} odd, the transfer zero at $\omega = \pi$ for type II linear-phase filters violates this condition.

To illustrate the advantages of the Oversampling+B-spline structure, the design example of the preceding section is adapted to this resampling function. The resulting continuous frequency response is depicted in Figure 5.10. The stopband attenuation of this design is 75.43 dB, which is an improvement of about 14.05 dB compared to the optimized Oversampling+Lagrange design with the same parameters. Due to the uniform weighting used, the passband error is decreased by the same ratio. As intended, the performance improvement is primarily gained by the improved image attenuation of the B-spline basis function compared to a Lagrange interpolator of the same order. Moreover, the passband detail shows how the magnitude roll-off, which is characteristic to B-spline basis functions, is compensated by the discrete-time filter $H_{dig}(e^{j\omega})$.

Parameters			Oversampling+Lagrange		Oversampling+B-spline		Improvement	
L	N_{int}	N_{dig}	$\delta_p^{(l)}$	$A_s^{(l)}$	$\delta_p^{(s)}$	$A_s^{(s)}$	$\dfrac{\delta_p^{(s)}}{\delta_p^{(l)}}$	$A_s^{(s)} - A_p^{(l)}$
4	3	63	$1.05 \cdot 10^{-3}$	59.54 dB	$3.38 \cdot 10^{-4}$	69.43 dB	32.03 %	9.89 dB
4	3	95	$8.54 \cdot 10^{-4}$	61.38 dB	$1.69 \cdot 10^{-4}$	75.43 dB	19.83 %	14.05 dB
4	3	127	$7.85 \cdot 10^{-4}$	62.10 dB	$1.46 \cdot 10^{-4}$	76.71 dB	18.62 %	14.60 dB
4	3	128	$7.83 \cdot 10^{-4}$	62.13 dB	$1.46 \cdot 10^{-4}$	76.71 dB	18.66 %	14.58 dB
4	3	191	$7.34 \cdot 10^{-4}$	62.69 dB	$1.31 \cdot 10^{-4}$	77.68 dB	17.79 %	14.99 dB
4	3	255	$7.17 \cdot 10^{-4}$	62.90 dB	$1.25 \cdot 10^{-4}$	78.11 dB	17.40 %	15.21 dB
4	4	255	$3.82 \cdot 10^{-4}$	68.37 dB	$1.45 \cdot 10^{-5}$	96.79 dB	3.79 %	28.41 dB
4	5	255	$5.86 \cdot 10^{-5}$	84.65 dB	$1.76 \cdot 10^{-6}$	115.11 dB	3.00 %	30.46 dB
4	6	255	$3.24 \cdot 10^{-5}$	89.80 dB	$2.27 \cdot 10^{-7}$	132.87 dB	0.70 %	43.07 dB
4	7	255	$5.53 \cdot 10^{-6}$	105.16 dB	$4.70 \cdot 10^{-8}$	148.05 dB	0.85 %	42.89 dB
2	5	255	$3.22 \cdot 10^{-3}$	49.84 dB	$1.60 \cdot 10^{-4}$	75.94 dB	4.95 %	26.10 dB
3	5	255	$3.02 \cdot 10^{-4}$	70.40 dB	$1.03 \cdot 10^{-5}$	99.78 dB	3.40 %	29.37 dB
4	5	255	$5.86 \cdot 10^{-5}$	84.65 dB	$1.76 \cdot 10^{-6}$	115.11 dB	3.00 %	30.46 dB
6	5	255	$6.74 \cdot 10^{-6}$	103.43 dB	$2.05 \cdot 10^{-7}$	133.81 dB	3.05 %	30.37 dB

Table 5.5: Design experiments for Oversampling+B-spline structures. Design with overall optimization scheme and L_∞ norm. Comparison to optimized Oversampling+Lagrange design. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$.

In Table 5.5, the design experiments performed for the Oversampling+Lagrange structure in Table 5.3 are repeated for the Oversampling+B-spline structure. It shows that a performance improvement is consistently gained compared to optimized Oversampling+Lagrange structures with the same parameters.

5.5.2 Optimal MOMS Functions

Although the use of B-spline basis functions enables a significant performance gain compared to Lagrange interpolators, it remains questionable whether the resulting structures are optimal. In particular, because the stopband error is typically dominated by the first passband image, the asymptotic image attenuation $O(\omega^{-[N+1]})$ of B-spline basis functions, corresponding to a regularity order of $N-1$, appears overly restrictive.

A conceptually similar argument is found in [BTU01], albeit the criteria of optimality considered there are profoundly different. In Section 5.3.4, resampling functions with support L and approximation order L has been introduced as the class of *maximal-order interpolation of minimal support* (MOMS) functions. Among these, the interpolation kernel minimizing the asymptotic constant c with respect to the L_2 norm is sought (5.84)

$$\|u(x) - f(x)\|_2 \le c \times T^L \times \|f(x)\|_2 \qquad \text{for } T \to 0.$$

As argued in Section 5.3.4, MOMS interpolation kernels are of the form (5.88), corresponding
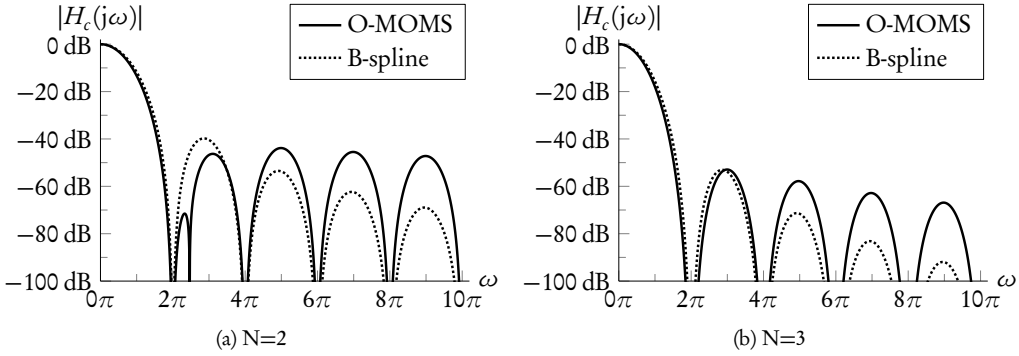
Figure 5.11: Continuous frequency response of O-MOMS functions [BTU01] compared to B-spline basis functions.

to a continuous frequency response

$$H_c(j\omega) = \text{sinc}\left(\frac{\omega}{2}\right)^{N+1} \sum_{n=0}^{N} j^n \gamma_n \omega^n . \tag{5.94}$$

Closed-form expressions for the coefficients γ_n that minimize the constant c are derived in [BTU01]. The resulting resampling filters are consequently termed *optimal MOMS* or *O-MOMS*.

In Figure 5.11, the frequency responses of O-MOMS functions of orders $N = 2$ and $N = 3$ are compared to B-spline basis functions of the same order. It is observed that the attenuation in the region around $\omega = 2\pi$, that is, at the location of the first passband image, is increased compared to the spline basis function. On the other hand, the attenuation of higher-order images as well as the rate of decay towards higher frequencies is lower than in case of spline basis functions.

A design example utilizing an O-MOMS interpolation kernel and the proposed overall optimization scheme is shown in Figure 5.12. Compared to the design based on spline basis functions, the minimum stopband attenuation A_s is further increased by about 10.4 dB to 85.82 dB. The maximum passband error is decreased by the same ratio. The frequency responses show that the improvement basically results from an increased stopband attenuation of $H_{int}(j\omega)$ in the region of the first passband image of $H_{dig}(e^{j\omega})$.

In Table 5.6, the design variations considered in the previous sections are repeated for Over-sampling+O-MOMS structures and compared to systems utilizing B-spline basis functions. It can be seen that a performance improvement is gained for all considered designs. However, as illustrated by the first three examples, a substantial improvement is achieved only if the order N_{dig} of the discrete-time prefilter $H_{dig}(e^{j\omega})$ exceeds a certain limit. Otherwise, the performance of the overall system is restricted by the stopband rejection of $H_{dig}(e^{j\omega})$ rather than by the image attenuation of $H_{int}(j\omega)$.

Despite the evident performance gains compared to Lagrange or B-spline interpolation kernels, it is important to reemphasize that the optimality criteria for O-MOMS functions are quite different from those relevant for resampling applications. First, the derivation of O-MOMS is tightly bound to the L_2 norm, while other norms such as L_∞ are important for ASRC

Figure 5.12: Design example for Oversampling+O-MOMS structure. Parameters $L = 4$, $N_{int} = 3$, $N_{dig} = 95$, $\omega_c = 0.75\pi$. Design with overall optimization scheme, L_∞ norm, $W(X) = 1$. Comparison to Oversampling+B-spline design.

applications as well. Second, the approximation order and the corresponding asymptotic constant c (5.84) are not directly related to performance measures used in ASRC filters, such as the passband deviation or the minimum stopband attenuation. Finally, oversampled input signals are not considered in [BTU01].

5.5.3 Fixed Resampling Functions with Optimized Image Band Attenuation (OIB)

As noted above, the resampling functions considered so far are not especially adapted to the characteristics of ASRC systems incorporating oversampling. However, it is possible to design continuous-time resampling functions that are specifically matched to this role.

Advantageous properties of resampling filters based on symmetric piecewise polynomial functions have been pointed out in Section 3.5, and the preceding sections further confirmed this argument. For this reason, the continuous-time resampling functions considered here are restricted to this class, which is equivalent to the class of resampling filters that can be implemented by the modified Farrow structure. Moreover, the order of the polynomial pieces is constrained to equal the subfilter order N of the Farrow structure, that is $M = N = N_{int}$, in the following. While this restriction is arbitrary to some extent, there are also objective reasons that support this choice. First, all specialized resampling functions considered so far, namely Lagrange interpolators, spline basis functions and O-MOMS functions, fulfill this equality $M = N$. An identical choice enables a direct comparison to these resampling filters. Second, as manifested in the concept of MOMS interpolation kernels (see Section 5.3.4), all resampling functions that maximize the approximation order in the L_2 sense for a given filter order N_{int} are piecewise polynomial functions with $M = N$. This justifies the special role of this class of functions.

In the preceding sections, it became apparent that the image attenuation of $H_{int}(j\omega)$ in the

Parameters			Oversampling+B-spline		Oversampling+O-MOMS		Improvement	
L	N_{int}	N_{dig}	$\delta_p^{(s)}$	$A_s^{(s)}$	$\delta_p^{(o)}$	$A_s^{(o)}$	$\dfrac{\delta_p^{(o)}}{\delta_p^{(s)}}$	$A_s^{(o)} - A_s^{(s)}$
4	3	63	$3.38 \cdot 10^{-4}$	69.43 dB	$3.35 \cdot 10^{-4}$	69.49 dB	99.31 %	0.06 dB
4	3	95	$1.69 \cdot 10^{-4}$	75.43 dB	$5.12 \cdot 10^{-5}$	85.82 dB	30.24 %	10.39 dB
4	3	127	$1.46 \cdot 10^{-4}$	76.71 dB	$4.04 \cdot 10^{-5}$	87.87 dB	27.63 %	11.17 dB
4	3	128	$1.46 \cdot 10^{-4}$	76.71 dB	$4.02 \cdot 10^{-5}$	87.93 dB	27.48 %	11.22 dB
4	3	191	$1.31 \cdot 10^{-4}$	77.68 dB	$3.32 \cdot 10^{-5}$	89.59 dB	25.41 %	11.90 dB
4	3	255	$1.25 \cdot 10^{-4}$	78.11 dB	$3.08 \cdot 10^{-5}$	90.27 dB	24.68 %	12.16 dB
4	4	255	$1.45 \cdot 10^{-5}$	96.79 dB	$2.81 \cdot 10^{-6}$	111.05 dB	19.40 %	14.26 dB
4	5	255	$1.76 \cdot 10^{-6}$	115.11 dB	$3.23 \cdot 10^{-7}$	129.81 dB	18.39 %	14.71 dB
4	6	255	$2.27 \cdot 10^{-7}$	132.87 dB	$4.06 \cdot 10^{-8}$	147.89 dB	17.84 %	15.02 dB
4	7	255	$4.70 \cdot 10^{-8}$	148.05 dB	$4.17 \cdot 10^{-8}$	154.23 dB	88.58 %	6.18 dB
2	5	255	$1.60 \cdot 10^{-4}$	75.94 dB	$5.14 \cdot 10^{-5}$	85.78 dB	32.22 %	9.84 dB
3	5	255	$1.03 \cdot 10^{-5}$	99.78 dB	$2.32 \cdot 10^{-6}$	112.68 dB	22.63 %	12.90 dB
4	5	255	$1.76 \cdot 10^{-6}$	115.11 dB	$3.23 \cdot 10^{-7}$	129.81 dB	18.39 %	14.71 dB
6	5	255	$2.05 \cdot 10^{-7}$	133.81 dB	$3.19 \cdot 10^{-8}$	149.93 dB	15.52 %	16.12 dB

Table 5.6: Design experiments for Oversampling+O-MOMS structures. Design with overall optimization scheme and L_∞ norm. Comparison to Oversampling+B-spline design. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$.

frequency regions $[2\pi k - \omega_c/L, 2\pi k + \omega_c/L]$, $k = \pm1, \pm2, \ldots$, is of paramount importance for the achievable performance of an ASRC system incorporating oversampling. Therefore, the design goal for the continuous-time resampling filter is to maximize the attenuation in the passband image regions of the discrete-time prefilter $H_{dig}(e^{j\omega})$. For this reason, this design is referred to as *optimized image bands* (OIB) in the following. Consequently, the resulting structures are termed *Oversampling+OIB*.

In contrast to image band errors, magnitude response errors in the passband region $|\omega| < \omega_c/L$ of $H_{int}(j\omega)$ can be corrected within certain limits by a suitable modification of the discrete-time prefilter. Thus, it appears at the first glance that $H_{int}(j\omega)$ can be designed based on the stopband specification only. However, a large magnitude of $H_{dig}(e^{j\omega})$ at a passband frequency to compensate a deviation caused by $H_{int}(j\omega)$ will also cause amplification at corresponding image frequencies, thus decreasing the stopband attenuation of the whole system. Therefore, it is necessary to place sensible conditions on the passband response of $H_{int}(j\omega)$ to prevent such degenerate solutions. At the same time, the constraints should not be too strict. Otherwise, the passband specifications would prevent the design method from achieving a good stopband attenuation. A sensible choice is to specify a maximum passband error δ_p, resulting in a constrained optimization problem.

The design process for the Oversampling+OIB structure consists of two steps. First, the continuous-time resampling filter is designed as a modified Farrow structure. In the second step, the frequency response of the obtained filter is used to design the discrete-time prefilter $H_{dig}(e^{j\omega})$ applying the overall optimization scheme introduced in Section 5.4.

The primary design goal for the continuous-time resampling filter $H_{int}(j\omega)$ is to maximize

the attenuation in the image regions of the passband of $H_{dig}(e^{j\omega})$. This requirement can be stated using an adaptation of the nonuniform stopband specification shown in Table 3.5

$$X_s = \bigcup_{k=1}^{\infty} [\pi k - \omega_c/L, 2\pi k + \omega_c/L].$$ (5.95)

In this way, optimization of $H_{int}(j\omega)$ according to the OIB criterion is stated as a design problem for the modified Farrow structure as described in Section 5.2.4, utilizing a nonuniform stopband specification subject to a fixed passband error constraint.

```
cvx_begin
    variable  b_int (N_coeffs);
    minimize ( norm(  G(X_s) b_int − Ĥ_c(X_s),  Inf );
    subject to abs ( G(X_p) b_int − Ĥ_c(X_p)) <= δ_p;
cvx_end
```

Here $\mathbf{G}(\mathbf{X}_p)$ and $\mathbf{G}(\mathbf{X}_s)$ refer to the partitions of the discrete transfer matrix $\mathbf{G}$ (5.40) corresponding to the discrete passband and stopband frequency grids, respectively. $\mathbf{b}_{int}$ denotes the coefficient vector (5.38) of the modified Farrow structure that parameterizes $H_{int}(j\omega)$.

Numerical test have shown that the obtained filters are relatively insensitive to the choice of the passband error limit. Thus, this value can be chosen relatively large, e.g. $\delta_p = 0.5$, because the passband error is compensated in the subsequent design of the discrete-time filter $H_{dig}(e^{j\omega})$. Notwithstanding, it proves useful to normalize the coefficients of $H_{int}(j\omega)$, for instance by restricting the static gain of the filter to unity

$$\mathbf{b}'_{int} = \frac{1}{H_{int}(0)}\mathbf{b}_{int}.$$

Among other advantages, this normalization eases a direct comparison to other resampling functions. It is noted that the above passband constraints are not unique, neither do they ensure an optimal quality of the overall ASRC system. However, the fixed passband error specification yields good results for a large variety of designs.

In Figure 5.13, the frequency response $H_{int}(j\omega)$ maximizing the image band attenuation for $N_{int} = 5$, $M = 5$, $L = 4$, $\omega_c = 0.75\pi$ is shown. In comparison with Lagrange interpolation, spline basis functions, and an O-MOMS kernel of the same order, the region of high image attenuation is broadened significantly and spans the complete region of the passband images of $H_{dig}(e^{j\omega})$, thus conforming to the design objective. On the other hand, the passband error is moderate and comparable to the O-MOMS function.

The second step comprises the design of the coefficients $b[k]$ of the discrete-time prefilter $H_{dig}(e^{j\omega})$. The overall frequency response of the Oversampling+OIB structure follows from (5.91a)

$$H_c(j\omega) = \sum_{k=0}^{N'_{dig}} b[k]G_o(k,\omega) \qquad \text{with}$$ (5.96a)
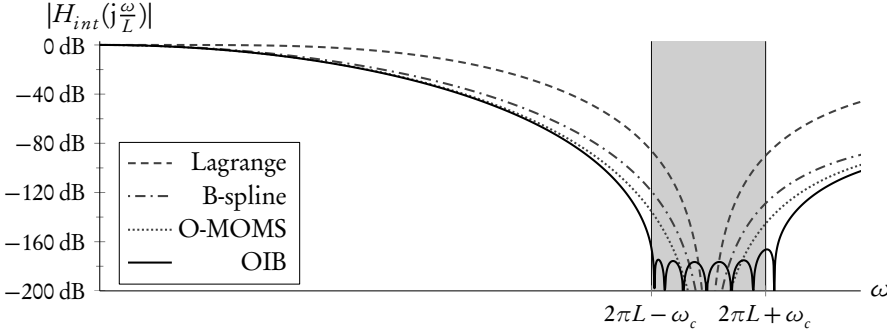
Figure 5.13: Frequency response of optimized image band attenuation (OIB) filter. Design parameters: $N_{int} = 5$, $M = 5$, $L = 4$, $\omega_c = 0.75\pi$. Comparison to other resampling functions.

$$G_o(k, \omega) = \frac{1}{L} \operatorname{trig}(k, \frac{\omega}{L}) \sum_{m=0}^{N_{int}} \sum_{n=0}^{N'_{int}} b_{mn} G\left(m, n, \frac{\omega}{L}\right), \qquad (5.96b)$$

with $G(m, n, \omega)$ denoting the basis functions (5.16) of the modified Farrow structure. Likewise, b_{mn} are the coefficients of the parameter vector $\mathbf{b}_{int}$ (5.38), and the index limit N'_{int} is given by (3.123). Using this representation, the overall optimization scheme introduced in Section 5.4 can be applied straightforwardly.

In Figure 5.14, the design example of the previous sections is repeated for the Oversampling+OIB structure. Compared to the Oversampling+O-MOMS design, the stopband error is decreased by a further 3.28 dB. In Table 5.7, the design examples used throughout this section are repeated for OIB resampling filters and compared to the results based on O-MOMS functions. The design quality is improved for nearly all parameter variations. However, for small prefilter orders such as $N_{dig} = 63$, the design error cannot be decreased further since it is dominated by the limited image attenuation of $H_{dig}(e^{j\omega})$.

For higher prefilter orders a different problem may occur. For instance, in the design example with $L = 4$, $N_{int} = 7$ and $N_{dig} = 255$, the stopband attenuation of the OIB design is decreased by 3.40 dB compared to the corresponding O-MOMS design. An analysis shows that the stopband attenuation of the OIB resampling filter in the passband image regions of $H_{dig}(e^{j\omega})$ is vastly better than the O-MOMS function. However, the overall performance is dominated by the attenuation in the frequency intervals corresponding to the images of the transition band of $H_{dig}(e^{j\omega})$. In these regions, the magnitude of the continuous-time filter $H_{int}(j\omega)$ cannot be sufficiently attenuated by means of a prefilter $H_{dig}(e^{j\omega})$ designed for this purpose.

This example reveals an inherent limitation of the proposed overall optimization scheme based on a fixed resampling filter. In the design of the continuous-time resampling function $H_{int}(j\omega)$, it is assumed that limitations of this component are ideally compensated by the prefilter $H_{dig}(e^{j\omega})$. However, if $H_{dig}(e^{j\omega})$ fails to achieve this compensation, the performance of the overall structure may become inferior to a more conventional design.
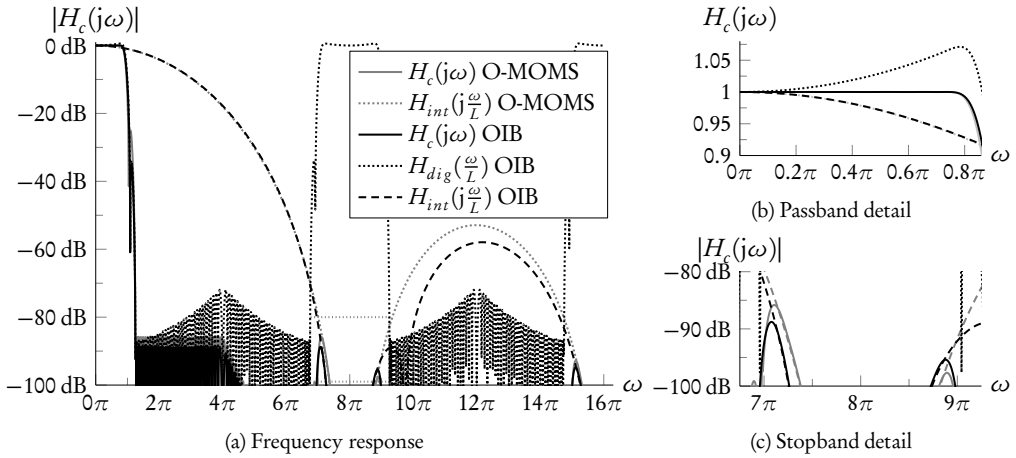
Figure 5.14: Design example for Oversampling+OIB structure. Parameters $L = 4$, $N_{int} = 3$, $N_{dig} = 95$, $\omega_c = 0.75\pi$. Design with overall optimization scheme, L_∞ norm, $W(X) = 1$. Comparison to Oversampling+O-MOMS design.

It is worth emphasizing that this problem cannot be resolved by changing the design specification for $H_{int}(j\omega)$. For instance, if the transition band images of $H_{dig}(e^{j\omega})$ are added to the stopband specification for $H_{int}(j\omega)$, the performance of the overall ASRC structure is no longer limited by the frequency response in these regions. However, this modification decreases the minimum stopband attenuation of the continuous-time resampling filter, consequently decreasing the achieved quality for the majority of designs.

5.6 Overall Optimization of Modified Farrow Structures with Oversampled Input

In the preceding sections, an overall optimization scheme for ASRC structures based on oversampling and fixed continuous-time resampling functions has been investigated. It became clear that the adaptation of the discrete-time prefilter to the characteristics of the resampling filter $H_{int}(j\omega)$ enables significant performance improvements over a conventional design of these components. However, although some of the fixed resampling filters were designed to anticipate the behavior of the respective other component, the design processes are performed independently. Thus, the filters are not specifically matched to the actual characteristics of their counterparts. It is therefore worthwhile to consider design methods that enable a joint optimization of both components, preferably resulting in globally optimal solutions.

As pointed out in Section 5.5.3, symmetric piecewise polynomial resampling functions, which can be implemented by the modified Farrow structure, form a reasonable and general model for the continuous-time resampling function. Such ASRC structures based on integer-ratio oversampling and symmetric piecewise polynomial basis functions are denoted as *Oversampling+Farrow* in the following.

Parameters			Oversampling+O-MOMS		Oversampling+OIB		Improvement	
L	N_{int}	N_{dig}	$\delta_p^{(o)}$	$A_s^{(o)}$	$\delta_p^{(oib)}$	$A_s^{(oib)}$	$\dfrac{\delta_p^{(oib)}}{\delta_p^{(o)}}$	$A_s^{(oib)}-A_s^{(o)}$
4	3	63	$3.35 \cdot 10^{-4}$	69.49 dB	$3.36 \cdot 10^{-4}$	69.48 dB	100.05 %	0.00 dB
4	3	95	$5.12 \cdot 10^{-5}$	85.81 dB	$3.51 \cdot 10^{-5}$	89.10 dB	68.61 %	3.28 dB
4	3	127	$4.04 \cdot 10^{-5}$	87.87 dB	$2.46 \cdot 10^{-5}$	92.16 dB	61.01 %	4.29 dB
4	3	128	$4.02 \cdot 10^{-5}$	87.91 dB	$2.44 \cdot 10^{-5}$	92.25 dB	60.81 %	4.34 dB
4	3	191	$3.34 \cdot 10^{-5}$	89.55 dB	$1.79 \cdot 10^{-5}$	95.08 dB	53.72 %	5.53 dB
4	3	255	$3.11 \cdot 10^{-5}$	90.25 dB	$1.48 \cdot 10^{-5}$	95.65 dB	47.53 %	5.40 dB
4	4	255	$2.43 \cdot 10^{-6}$	110.34 dB	$6.30 \cdot 10^{-7}$	124.03 dB	25.89 %	13.69 dB
4	5	255	$3.23 \cdot 10^{-7}$	129.58 dB	$4.85 \cdot 10^{-8}$	146.27 dB	15.00 %	16.69 dB
4	6	255	$4.10 \cdot 10^{-8}$	147.88 dB	$3.17 \cdot 10^{-8}$	149.51 dB	77.32 %	1.62 dB
4	7	255	$3.27 \cdot 10^{-8}$	153.98 dB	$2.77 \cdot 10^{-8}$	150.57 dB	84.52 %	−3.40 dB
2	5	255	$5.18 \cdot 10^{-5}$	85.76 dB	$1.42 \cdot 10^{-6}$	115.13 dB	2.73 %	29.37 dB
3	5	255	$2.02 \cdot 10^{-6}$	112.03 dB	$1.93 \cdot 10^{-7}$	134.42 dB	9.53 %	22.39 dB
4	5	255	$3.23 \cdot 10^{-7}$	129.58 dB	$4.85 \cdot 10^{-8}$	146.27 dB	15.00 %	16.69 dB
6	5	255	$3.24 \cdot 10^{-8}$	149.93 dB	$1.27 \cdot 10^{-8}$	157.76 dB	39.28 %	7.83 dB

Table 5.7: Design experiments for Oversampling+OIB structures. Design with overall optimization scheme and L_∞ norm. Comparison to optimized Oversampling+O-MOMS design. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$.

5.6.1 Optimization Problem

For this class of continuous-time resampling functions, the frequency response $H_{int}(\mathrm{j}\omega)$ is determined by (5.6b) and is parameterized by a set of coefficients b_{mn} (3.130), corresponding to the elements of the coefficient matrix B of a modified Farrow structure. In this way, the overall frequency response of the Oversampling+Farrow structure is identical to that of an Oversampling+OIB design (5.96)

$$H_c(\mathrm{j}\omega) = \sum_{k=0}^{N'_{dig}} b[k] \frac{1}{L} \operatorname{trig}\left(k, \frac{\omega}{L}\right) \sum_{m=0}^{M} \sum_{n=0}^{N'} b_{mn} G(m,n,\omega). \tag{5.97}$$

Here, $\operatorname{trig}(n,\omega)$ (5.90c) and $G(m,n,\omega)$ (5.16) denote the basis functions of the discrete-time prefilter and of the modified Farrow structure, respectively.

It is apparent that the coefficients $b[k]$ of the discrete-time prefilter and those of the continuous-time resampling function, b_{mn} are combined in a multiplicative way. Consequently, the optimization problem resulting from (5.97) is non-convex and cannot be transformed into a convex problem straightforwardly. Indeed, the exclusion of expressions containing products of optimization variables forms a basic principle of convex optimization [Gra04, GB08, WBV98].

Two basic consequences follow from this characteristic. First, the large knowledge base of algorithms for convex optimization, which enable even large-scale problems to be solved with acceptable costs, cannot be used. Second, it is undecidable in general whether a solution obtained from a non-convex optimization method is globally optimal or represents only a local optimum.

(a) Original structure (b) Transformed structure
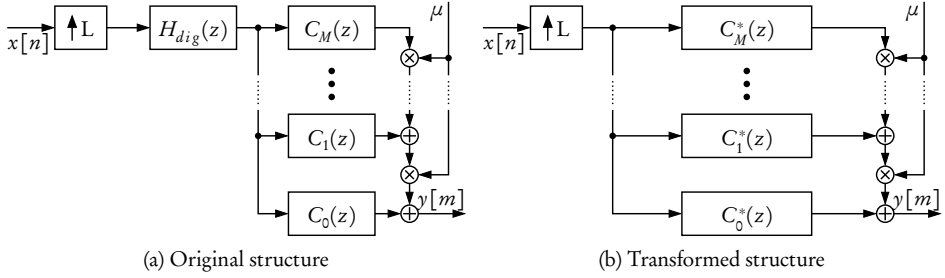
Figure 5.15: Transformation of the Oversampling+Farrow structure into a more general, convex structure.

5.6.2 Upper Performance Limit

Because a direct global optimization is apparently not feasible, it is desirable to obtain realistic estimates of the maximally achievable performance. Indeed, such an upper limit can be determined by a transformation of the structure. In Figure 5.15a, an ASRC system consisting of oversampling and a modified Farrow structure with subfilters responses $C_m(z)$ is shown. Because both $H_{dig}(z)$ and the subfilters $C_m(z)$ are discrete-time linear-phase filters, they can be combined into $M + 1$ discrete-time linear-phase filters $C_m^*(z)$ of order $N^* = N_{int} + N_{dig}$. Their noncausal frequency response is determined by

$$C_m^*(\omega) = H_{dig}(\omega)C_m(\omega) \qquad \text{for } m = 0,\ldots,M \qquad (5.98a)$$

$$= \sum_{n=0}^{N^{*'}} b_{mn}^* \operatorname{trig}(m,n,\omega) \qquad \text{with } N^{*'} = \left\lfloor \frac{N_{int} + N_{dig}}{2} \right\rfloor, \qquad (5.98b)$$

where $\operatorname{trig}(m,n,\omega)$ denote the trigonometric basis functions of symmetric piecewise polynomial functions (3.127). The coefficients b_{mn}^* are related to the coefficients $b[k]$ and b_{mn} by a discrete-time convolution. Based on these subfilter responses, the continuous frequency response of the complete structure follows from (5.18)

$$H_c(\mathrm{j}\omega) = \frac{T_i}{L} \sum_{m=0}^{M} \Phi(m,\frac{\omega}{L})C_m^*\left(\frac{\omega}{L}\right). \qquad (5.99)$$

The scaling factor $\frac{1}{L}$ that is applied both to the magnitude and the frequency variable ω follows from the sampling rate expansion (see Section 3.4.2). Aside from this scaling, (5.99) represents the frequency response of a symmetric piecewise polynomial resampling filter. Therefore, the design methods for the modified Farrow structure described in Section 5.2.4 can be applied directly. Because these corresponding design problems are convex, these methods yield globally optimal results.

The class of ASRC systems represented by (5.99) is more general than the class of Oversampling+Farrow structures and includes the latter class as a subset. For this reason, the optimal objective value for the oversampled Farrow structure based on the subfilters $C_m^*(z)$ forms an

(a) Subfilters in commutator form   (b) Generalized Farrow structure form
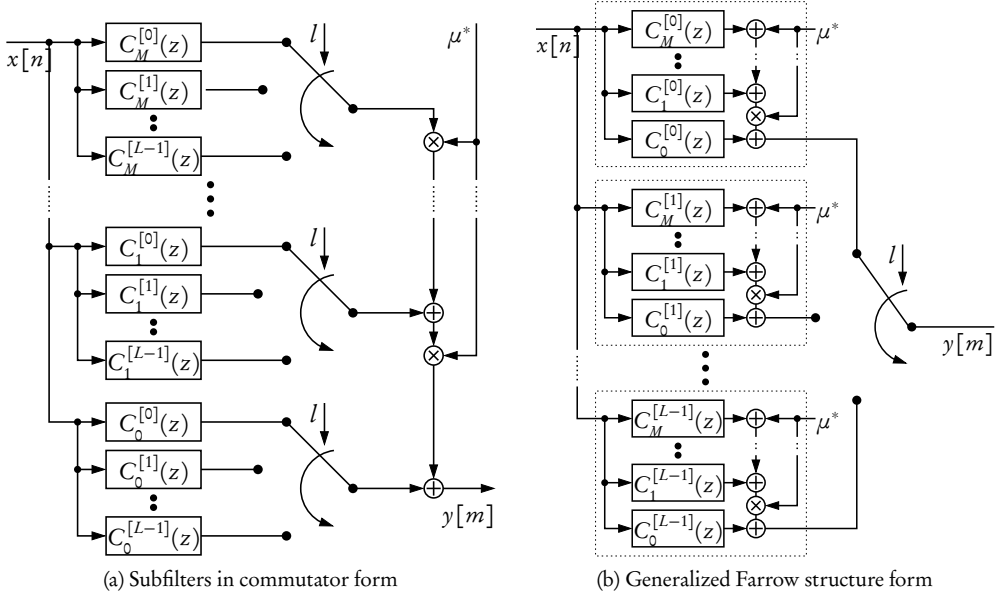
Figure 5.16: Equivalence between the transformed structure for Oversampling+Farrow and the generalized Farrow structure.

upper bound for the performance of the corresponding Oversampling+Farrow structure that is not necessarily reached by any element of this subclass.

5.6.3 Relation to the Generalized Farrow Structure

Although the structure shown in Figure 5.15b resulted from a transformation of an Oversampling+Farrow structure, it is actually identical to the generalized Farrow structure described in Section 3.6.6. This equivalence can be shown using network transformations and polyphase decompositions as used in the theory of multirate systems, e.g. [CR83, Vai92].

In the transformed structure according to Figure 5.15b, the sample rate expander can be transposed into each subfilter branch without altering the behavior of the system. In this way, each subfilter branch represents a system for a 1-to-L sample rate increase as described in Section 3.4.2. These branches can be represented in commutator form based on a polyphase decomposition of the filters $C_m^*(z)$. The corresponding system is shown in Figure 5.16a. The polyphase filters $C_m^{[l]}(z)$, $l = 0, 1, \ldots, L-1$ are related to the subfilters $C_m^*(z)$ by

$$C_m^{[l]}(z) = \sum_{n=0}^{\frac{N^*+1}{L}-1} c_{m(nL+l)}^* z^{-n} \qquad \text{with} \quad C_m^*(z) = \sum_{n=0}^{N^*} c_{mn}^* z^{-l}. \tag{5.100}$$

It is assumed that the length of the filters $C_m^*(z)$, i.e. $N^* + 1$, is an integer multiple of L, which can be guaranteed by an appropriate zero padding without loss of generality.

The relative order between the commutators and the polynomial evaluation can be interchanged using network identities, resulting in the system depicted in Figure 5.16b. Here, the transformed structure is represented by L parallel Farrow structures. An output sample $y[m]$ is selected from one of these structures depending on an index l which is determined by the intersample position. This structure is identical to the commutator form of the generalized Farrow structure shown in Figure 3.36b of Section 3.6.6. In this way, the proposed structure obtained by transforming an Oversampling+Farrow structure is equivalent to the generalized Farrow structure, where the subfilters $C_m^{[l]}(z)$ are formed by polyphase decomposition of the subfilters $C_m^*(z)$.

This close relation between the generalized Farrow structure and Oversampling+Farrow systems gives rise to several interesting conclusions. First, it must be noted that this relation does not come unexpectedly, as the generalized Farrow structure has been initially proposed in the context of structures involving oversampling [Ram98]. Second, the generalized Farrow structure can be considered as an Oversampling+Farrow structure where the functionality of the discrete-time filter $H_{dig}(z)$, that is, the attenuation of the signal images caused by the sample rate expander, is incorporated into each subfilter $C_m^*(z)$. It appears worthwhile to investigate whether this additional degree of freedom, that is, to have an individually designed anti-imaging filter for each subfilter branch, results in a significantly improved resampling quality compared to conventional Oversampling+Farrow structures.

Finally, this equivalence can be used to perform coefficient design for the generalized Farrow structure. Design methods for this structure are not covered extensively in literature. However, by combining the filters $C_m^{[l]}(z)$ into the subfilters $C_m^*(z)$, the frequency response of the generalized Farrow structure is equivalent to that of a Farrow structure subject to a scaled frequency axis. If the continuous impulse response of the generalized Farrow structure fulfills the symmetry condition motivated in Section 3.5.4, then the continuous frequency response is given by (5.99). In this case, the efficient design methods for the modified Farrow structure are directly applicable to the generalized Farrow structure.

5.6.4 Iterative Optimization Procedure

While the relation to the generalized Farrow structure establishes an upper performance limit, it does not provide a means to design the filter coefficients for an Oversampling+Farrow structure. A practical approach to this non-convex design problem is to optimize the components of this structure iteratively. Such methods have been applied, for instance, in [VS07] or, in the context of VFD filters, in [JL03].

In each iteration, the coefficient vectors parameterizing $H_{dig}(e^{j\omega})$ and $H_{int}(j\omega)$ are designed sequentially using the proposed overall optimization scheme. For each design problem, the transfer matrix of the respective other component is determined using the most recent coefficient set for this filter. This algorithm is best illustrated by a pseudocode listing formulated in CVX.

```
b_int = <initial value>;
while not( terminate )
    H_int = G_int    b_int;
    cvx_begin
        variable b_dig (N_dig +1)
```

```
        minimize ( norm ( W . H_int . ( G_dig  b_dig −Ĥ_c , Inf ) ;
        subject to <constraints >;
    cvx_end
    H_dig = G_dig    b_dig ;
    cvx_begin
        variable b_int (N_coeffs)
        minimize ( norm ( W . H_dig . ( G_int  b_int −Ĥ_c , Inf ) ;
        subject to <constraints >;
    cvx_end
end
```

Here, $\mathbf{G}_{dig}$ denotes the discrete transfer matrix for the prefilter $H_{dig}(z)$ (5.92b), and $\mathbf{b}_{dig}$ represents the corresponding coefficient vector. The transfer matrix G_{int} for the continuous-time resampling function $H_{int}(j\omega)$ follows from the matrix for the modified Farrow structure (5.40), but is adapted to account for the frequency scaling due to the oversampled input

$$
\mathbf{G}_{int} = \frac{1}{L}
\begin{bmatrix}
G(0,0,\frac{\omega_0}{L}) & \cdots & G(0,N',\frac{\omega_0}{L}) & \cdots & G(M,0,\frac{\omega_0}{L}) & \cdots & G(M,N',\frac{\omega_0}{L}) \\
G(0,0,\frac{\omega_1}{L}) & \cdots & G(0,N',\frac{\omega_1}{L}) & \cdots & G(M,0,\frac{\omega_1}{L}) & \cdots & G(M,N',\frac{\omega_1}{L}) \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\
G(0,0,\frac{\omega_K}{L}) & \cdots & G(0,N',\frac{\omega_K}{L}) & \cdots & G(M,0,\frac{\omega_K}{L}) & \cdots & G(M,N',\frac{\omega_K}{L})
\end{bmatrix} .
$$

Likewise, $\mathbf{b}_{int} = \{b_{mn}\}$ (5.38) represents the coefficient vector for $H_{int}(j\omega)$.

The termination condition can be specified in several ways. One possibility is to set a fixed number of iterations. Alternatively, the algorithm can be controlled by conditions on the obtained solutions. For instance, the termination might be triggered if the change of the coefficient vectors between successive iterations falls below a predefined margin, indicating the convergence of the algorithm.

As with general non-convex optimization problems, the quality of the solution depends on the initial parameter vector provided to the solution algorithms. There are two principal ways to determine this initial solution. First, the algorithm may start with an initial value for $\mathbf{b}_{int}$ as shown in the above algorithm. For instance, any of the fixed resampling filters characterized in the preceding sections can be used to determine these coefficients. Second, the algorithm may start from an initial coefficient set for the prefilter $H_{dig}(z)$. In this case, the ordering of the design problems within the main loop of the algorithm must be interchanged. The initial coefficient vector $\mathbf{b}_{dig}$ can be obtained by a conventional lowpass filter design for a rational sample rate converter.

5.6.5 Results and Discussion

To assess the improvements achieved by the joint optimization scheme, the iterative design method is applied to the example employed in the preceding sections. The optimization starts from an initial solution for the continuous-time resampling function. Specifically, the initial coefficients $\mathbf{b}_{int}$ are determined according to the optimized image band (OIB) design proposed in Section 5.5.3. The iterative optimization process is performed for ten iterations and shows a monotonically decreasing and converging design error. As seen in Figure 5.17, the minimum
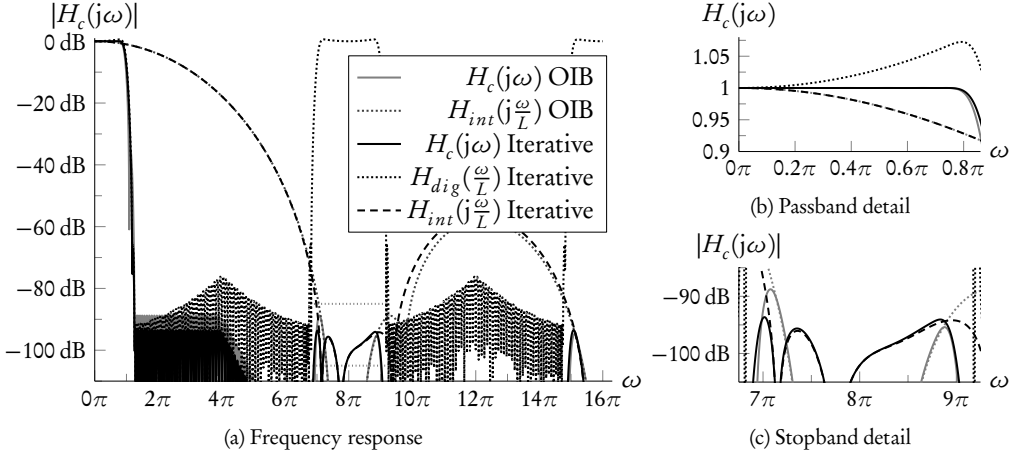
Figure 5.17: Design example for Oversampling+Farrow structure. Parameters $L = 4$, $N_{int} = 3$, $N_{dig} = 95$, $\omega_c = 0.75\pi$. Joint design with iterative optimization scheme, L_∞ norm, $W(X) = 1$. OIB design as initial value for $H_{int}(j\omega)$. Comparison to Oversampling+OIB design.

stopband attenuation A_s is increased to 93.42 dB, which is an improvement of about 5.14 dB compared to a design using a fixed OIB resampling filter, which represents the best solution so far.

In the stopband detail shown in Figure 5.17c, it is observed that this improvement basically results from an increased attenuation in the images of the transition regions of $H_{dig}(e^{j\omega})$. In particular, the frequency range $6.75\pi < \omega < 7.25\pi$ represents the first transition band image of $H_{dig}(e^{j\omega})$. As argued in Section 5.5.3, the attenuation in this region limits the performance of the whole structure in case of a fixed OIB resampling filter. In contrast, the proposed iterative optimization increases the attenuation in this region by adjusting the frequency responses of both filters. Evidently, this adaptation is possible only if $H_{dig}(e^{j\omega})$ and $H_{int}(j\omega)$ are jointly optimized. This affirms the usefulness of the iterative optimization procedure.

The upper limit for the achievable performance, obtained from a design for the corresponding generalized Farrow structure as described above, amounts to $A_s = 93.97$ dB. Thus, the stopband attenuation of the coefficients obtained from the iterative joint optimization procedure is very close to the theoretical upper performance bound.

For further performance assessment, the iterative optimization procedure is applied to the design examples used throughout this chapter. In Table 5.8, the achieved minimum stopband attenuation is presented for different initial solutions. In addition, this table shows the stopband attenuation for conventional Oversampling+Lagrange designs as well as for the overall optimization scheme with fixed resampling functions as proposed in the preceding sections. For this reason, Table 5.8 also summarizes the performance improvements for ASRC structures based on oversampling that have been proposed in this chapter.

It is observed that for virtually all parameter combinations and initial solutions, the iterative optimization scheme gains an improved design quality compared to the respective starting value.

Parameters			Conventional Lagrange	Fixed Resampling Functions				Iterative Optimization					Upper bound
L	N_{int}	N_{dig}		Lagrange	B-spline	O-MOMS	OIB	Lowpass	Lagrange	B-spline	O-MOMS	OIB	
4	3	63	57.86	59.54	69.43	69.49	69.48	65.52	68.36	69.43	69.49	69.48	72.30
4	3	95	57.34	61.38	75.43	85.82	88.28	75.82	72.18	85.42	92.33	93.42	93.97
4	3	127	56.97	62.10	76.70	87.88	91.40	74.91	77.03	87.76	93.66	94.83	95.88
4	3	128	56.97	62.12	76.70	87.92	91.49	73.23	78.59	87.77	93.87	94.73	95.95
4	3	191	56.53	62.69	77.68	89.58	94.49	77.65	76.13	89.90	94.91	96.55	97.19
4	3	255	60.02	62.89	78.10	90.26	95.32	74.64	77.66	91.62	97.76	97.35	97.70 [a,b]
4	4	191	60.58	68.02	95.95	109.62	120.11	90.64	77.76	111.48	119.71	128.15	129.31
4	5	191	75.55	83.84	113.51	127.75	139.84	108.97	108.10	125.87	146.79	149.54	157.19
4	6	191	79.45	88.71	130.65	144.51	145.14	119.47	93.15	154.91	155.03	152.31	172.13 [a]
4	7	191	93.97	103.56	146.77	159.95	150.71	134.30	119.10	160.84	161.34	161.65	178.74
2	5	191	47.08	49.79	75.66	85.31	113.61	74.08	76.66	94.96	100.87	122.22	126.59
3	5	191	64.91	70.06	98.94	111.51	130.37	99.59	95.23	116.15	133.33	139.50	145.35
4	5	191	75.55	83.84	113.51	127.75	139.84	108.97	108.10	125.87	146.79	149.54	157.19
6	5	191	97.37	101.16	128.08	128.08	128.11	124.15	108.75	128.08	128.11	128.11	130.10

Table 5.8: Design experiments for Oversampling+Farrow with iterative optimization. Design according to L_∞ norm, iterative optimization procedure with different initial values. Comparison to conventional design and the overall optimization scheme with fixed resampling functions. Fixed parameters: $\omega_c = 0.75\pi$, $W(X) = 1$.

[a] Performance of iterative optimization starting from O-MOMS design exceeds OIB as initial value, see text
[b] Performance of iterative design starting from O-MOMS solution slightly exceeds upper performance bound due to numerical inaccuracies, see text.

Moreover, the performance of the jointly optimized designs depends strongly on the quality of the initial solution. Generally, a good performance of the initial solution implies a good quality of the design generated by the iterative optimization scheme. Thus, the design process for the Oversampling+Farrow structure benefits from the fixed resampling functions introduced in Section 5.5. In most cases, the OIB design results in the best quality, which correlates to its good performance when used as a fixed resampling filter. However, there are exceptions to this rule. For the parameter combinations $L = 4$, $N_{int} = 3$, $N_{dig} = 255$ or $L = 4$, $N_{int} = 6$, $N_{dig} = 191$, the performance of the jointly optimized system starting from an O-MOMS function exceeds that of a design starting from an OIB filter. Basically, this behavior emphasizes the non-global nature of the non-convex iterative optimization scheme.

The theoretical performance bound for these design examples, obtained from the corresponding generalized Farrow structure, is shown in the rightmost column of Table 5.8. It can be seen that in most cases, the best designs obtained by the joint optimization scheme come very close to this theoretical optimum. In particular, this applies to the design examples with stopband attenuations up to about 130 dB, which appear to be most relevant for practical application. Again, it should be reemphasized that this upper bound corresponds to a more general structure and is not necessarily reached by any element of the class of Oversampling+Farrow structures considered.

Furthermore, it has to be noted that the performance figures are obtained from numerical optimization methods and thus exhibit only a finite precision. For instance, in the design example $L = 4$, $N_{int} = 3$, $N_{dig} = 255$, the stopband attenuation for the O-MOMS design marginally exceeds the upper performance bound by 0.06 dB. Analyses show that this seeming contradiction is a result of numerical effects, mainly due to the discretization of the frequency variable. So, the most likely interpretation of this result is that the iterative design procedure starting from the O-MOMS solution reaches the upper performance bound within the accuracy of the numerical optimization method.

As a byproduct, these investigations enable a performance comparison between Oversampling+Farrow structures and the generalized Farrow structure. As stated above, the quality achieved by an Oversampling+Farrow structure with oversampling ratio L and filter orders N_{dig} and N_{int} is typically only slightly inferior to that of a generalized Farrow structure with L parallel Farrow coefficient sets with orders $M = N_{int}$ and $N = N_{dig} + N_{int}$. Consequently, it appears unlikely that the additional effort in terms of instructions, coefficient storage and control logic required by the generalized Farrow structure pays off compared to a properly designed Oversampling+Farrow structure.

5.7 Conclusions

In this chapter, several improvements to ASRC algorithms and implementation structures have been proposed. A main focus has been placed on analytic descriptions for the continuous frequency response for these structures. Specifically, concise formulas for symmetric piecewise polynomial functions, which can be implemented by the modified Farrow structure, as well as for Lagrange interpolation and for systems based on oversampling and continuous-time resampling functions have been derived. These representations do not only facilitate design and analysis, but also enable insight into the characteristics of the corresponding structures.

A second focus is on coefficient design for these structures. In particular, purposeful design methods that enable globally optimal performance with respect to a selectable error norm and optional, additional design constraints are targeted. It is demonstrated that the framework of convex optimization is well-suited to fulfill these requirements and provides an efficient means to solve the corresponding optimization problems.

ASRC structures consisting of integer-ratio oversampling and continuous-time resampling functions such as Lagrange interpolation consist of two separate components, which are typically designed independently. Based on the frequency response of the complete system, an overall optimization scheme is proposed that enables performance improvements for virtually all design parameters. However, the ability to perform purposeful, optimal coefficient design with respect to a prescribed design specification is the main advancement of this optimization scheme.

However, it becomes obvious that widely used resampling functions such as Lagrange interpolation show severe shortcomings when used in combination with oversampling. A set of fixed resampling filters, ranging from spline basis functions to specifically designed filters, is investigated. In conjunction with the proposed overall optimization scheme, these functions enable a significant performance improvement for ASRC systems incorporating oversampling.

Oversampling+Farrow structures, consisting of oversampling and a fully parameterizable modified Farrow structure, form a very general class of ASRC algorithms that allow for high performance. However, the corresponding coefficient design problem is non-convex. Thus, it cannot be directly solved efficiently or in a globally optimal way. Two approaches are proposed to tackle this problem. First, an optimization scheme is investigated that approximates the non-convex optimization problem by iteratively designing the two components in an alternating fashion. It is shown that this method yields significant performance improvements over designs with fixed resampling functions. However, the achieved performance strongly depends on the quality of the initial design used as starting value for the iterative design. In this way, the fixed resampling functions developed in this chapter can also be used advantageously for the more general class of Oversampling+Farrow structures. Second, a transformation of this structure enables an upper performance limit to be obtained. It is shown that the designs obtained from the iterative optimization method come very close to this theoretical performance limit for most of the design examples if the optimization is started from suitable initial solutions.

Chapter 6

Evaluation

6.1 Introduction

In the preceding chapters, different implementation structures for ASRC systems as well as several improvements to these structures have been proposed. However, partly due to this multitude of algorithms, it is difficult to select a structure that is optimally suited for a given application. The numerous design parameters offered by most implementation structures further complicate this choice.

This chapter aims at providing practical guidance for choosing an appropriate implementation structure and the corresponding design parameters such that a given design objective is satisfied with minimal computational cost. For that purpose, two kinds of analyses are performed. In Section 6.3, the influence of various design parameters and objectives on the achievable quality of ASRC structures, namely the modified Farrow structure and structures incorporating oversampling, is examined. Section 6.4 investigates the minimum computational cost of different implementation structures as a function of the desired resampling quality. In combination, these analyses facilitate the selection of an implementation structure suitable for a given application.

The performance analyses performed in this chapter are applicable to general-purpose ARSC applications. However, specific properties of the signal processing algorithms for WFS, namely operations on arbitrary-access structures for input data or the generation of a multitude of resampled signals from a single input signal, are supported by the computational model used in the performance analysis. In this way, the results are directly applicable to the WFS signal processing structures proposed in Chapter 4.

6.2 Computational Model for Performance Analysis

As argued earlier, its flexibility, for instance the support of variable conversion ratios, forms one of the main advantages of ASRC. This flexibility, however, also puts specific demands on the methods for performance analysis in order to be applicable for a wide range of applications. For this reason, the approach used in this chapter, including assumptions and limitations, is described in the following.

6.2.1 Evaluation Scenario

The ability of ASRC to provide conversion by variable ratios influences the performance analysis in several ways. First, the continuous frequency response $H_c(j\omega)$ is used for the characterization of the resampling quality due to its independence of the conversion ratio. However, as argued in Section 3.4.5, this independence holds only if the sample rate does not decrease. Therefore,

the performance evaluations performed here are effectively restricted to conversion ratios $R \geq 1$. While it is possible to extend the analysis to sample rate decreases, for instance by using transposed structures, this is outside the scope and the intended applications of this work. Finally, the conversion ratio affects the computational cost of the implementation structures. This interrelation is reflected in the metrics for the computational effort introduced in the next section.

Due to the exceedingly large number of combinations of design objectives and parameters, an exhaustive exploration of the design space is infeasible. While an effort is made to provide a general overview of the impact of these variables, most of the parameter variations aim at design specifications suitable for general resampling applications in digital audio. In particular, a minimum stopband attenuation of about $90 \dots 120$ dB, as established e.g. in [LK81a, Lag82, ZB94] as well as normalized angular cutoff frequencies in the range $[0.75\pi, 0.9\pi]$, covering audio signal bandwidths in the range $18 \dots 20$ KHz for common audio sampling frequencies such as $f_s = 44.1$ KHz or $f_s = 48$ KHz, are targeted. Moreover, fixed tolerances for the passband error, which are sensible for many audio applications, are investigated.

6.2.2 Metrics for the Computational Effort

Arithmetic Complexity   As customary in digital signal processing, the number of arithmetic operations is the primary measure for the computational cost of an ASRC implementation structure. As noted above, the computational effort of SRC algorithms depends both on the input and the output sampling frequency. Therefore, the addition rate R^+ and the multiplication rate R^* are introduced as the number of respective operations per seconds in the style of [CR81, CR83]

$$R^+ = N_i^+ f_i + N_o^+ f_o \tag{6.1a}$$
$$R^* = N_i^* f_i + N_o^* f_o. \tag{6.1b}$$

Here, N_i^+, N_i^*, N_o^+ and N_o^* represent the number of additions or multiplications per input or output sample, respectively. Likewise, f_i and f_o denote the input and output sampling frequencies. Based on these measures, the total instruction rate (measured in arithmetic operations per second) is established as

$$R^{total} = R^+ + R^* = N_i f_i + N_o f_o, \tag{6.2}$$

where $N_i = N_i^+ + N_i^*$ and $N_o = N_o^+ + N_o^*$ denote the number or arithmetic operations per input or output sample, respectively. The dependence on the input and output sampling frequency is adapted from [VS07], where the term multiplication rate is introduced as the number of operations per input sample and is given in the form $N_i + \beta \cdot N_o$, $\beta = f_o/f_i$ being the conversion ratio (3.52).

Because the computational efforts to process input or output samples are represented separately, the performance measures (6.1) are easily adapted to cases where a single input signal is used to create n individually resampled output signals. In this case, the addition, multiplication and instruction rates are given by

$$R^+ = N_i^+ f_i + N_o^+ f_o n \tag{6.3a}$$

$$R^* = N_i^* f_i + n \cdot N_o^* f_o \tag{6.3b}$$

$$R^{total} = \left(N_i^+ + N_i^*\right) f_i + \left(N_o^+ + N_o^*\right) n \cdot f_o . \tag{6.3c}$$

In this way, the results are directly applicable to the algorithm partitioning scheme for WFS proposed in Section 4.3.

However, on real hardware platforms, the number of operations is not the only measure relevant to actual algorithm performance. While the performance of an algorithm typically depends on the characteristics of the specific platform as well as on the implementation, several abstract performance measures that are independent of a particular target platform are defined here.

Data Memory Requirements   In the model described above, the processing of input samples is decoupled from the calculation of required output samples. Consequently, apart from storage for intermediate results of arithmetic operations or state vectors of digital filters, data memory is required to store the results of the processing of input samples. The amount of memory depends on the algorithm.

This view is compatible with the signal processing schemes for WFS developed in Section 4.3. There, the results of the preprocessing stage, which operates on the input signal, are stored in a delay-line-like structure enabling arbitrary access. The size of this data structure depends linearly on the size of the preprocessed data for each input sample and dominates the memory requirements for WFS signal processing in most applications. As stated there, the proposed algorithm partitioning might be considered as a space-time trade-off that gains a significant reduction in complexity by increasing the memory requirements for the delay line by a bounded factor. However, on some target platforms, for instance DSP or FPGA architectures, memory size might prove to be a limiting parameter. For this reason, the data memory requirements, expressed as number of values to be stored per input sample, are included as a measure for the computational complexity. It may be utilized, for instance, for trade-off analyses with respect to other parameters such as the resampling quality or the arithmetic complexity.

Memory Bandwidth per Output Sample   In addition the data memory size, the bandwidth required to access this memory affects the performance of an algorithm in several ways. First, the amount of data required for the calculation of each output sample typically affects the number of memory load operations, thus increasing the number of instructions required. Second, memory bandwidth may prove a limiting factor on numerous target architectures.

Here, the analysis is constrained to the memory accesses required for the calculation of output samples. This restriction is justified mainly by three reasons. First, the memory bandwidth used in storing the results of the processing of the input samples is proportional to the size of the intermediate data. Therefore, this metric is already covered by the data memory requirements measure introduced above. Second, the processing of the input signal typically consists of filtering operations that operate on local data and coefficient memory. The costs of operations on these memories depend highly on the target platform and specific implementation details. It is therefore difficult to include these accesses in a platform-independent performance measure.

Third, the bandwidth required for the output computation is most important for applications such as WFS. On the one hand, the delay-line-like structure holding the processed input samples is often stored in other levels of the memory hierarchy, for instance in external RAM memory in

case of DSP architectures. Therefore, these memory accesses might prove particularly costly, and bandwidth restrictions of the architecture may limit the overall performance. On the other hand, if multiple output signals are generated from each preprocessed input signal, then the memory accesses for computing the output samples are likely to dominate the bandwidth requirements of the complete algorithm.

The memory bandwidth is given as the number of data values that is accessed in the computation of a single output sample.

Implementation Latency   Virtually all ASRC algorithms are internally based on discrete-time filtering. Therefore, they generally introduce an implementation delay to allow the filters to be implemented in causal form. The amount of delay depends on the algorithms and the design parameters used.

Ultimately, the latency requirements are determined by the application. For instance, implementation delays are critical in live scenarios or if the ASRC algorithm is contained in a feedback loop. For other applications, such as the reproduction of recorded audio scenes, this delay is often negligible or is easily compensated in other parts of the signal processing system.

To assess the suitability for a particular application and to enable trade-offs against other design objectives, the implementation delay is included in the analysis. It is typically given as a number of samples.

6.3 Performance Analysis

Most implementation structures for ASRC offer several design parameters such as filter orders or oversampling ratios. In addition, methods for purposeful coefficient design for these structures typically permit a multitude of design specifications, such as error norms, the cutoff frequency, or specific passband or stopband error specifications, to be determined. In this section, several investigations are performed to quantify the effects of these parameters and specifications on the achievable design quality.

Of the several implementation structures characterized in Section 3.6, the modified Farrow structure and structures incorporating oversampling and polynomial-based resampling filters are investigated in detail. This restriction is justified by the following reasons: First, resampling functions with explicitly calculated filter coefficients such as Lagrange or spline interpolators typically fail to achieve acceptable resampling quality for wideband signals with respect to objective error norms unless they are combined with oversampling.

The transposed Farrow structure is applicable only for sample rate reductions and thus outside the scope of the evaluations performed here. As shown in Section 5.6.3, the performance of the generalized Farrow structure is most often only slightly superior to that of an Oversampling+Farrow structure with significantly lower complexity. Among the remaining variants of the Farrow structure, only the modified Farrow structure is considered here due to the reduced computational effort and its advantageous impulse response symmetry. For the same reasons, structures incorporating oversampling are confined to those with symmetric piecewise polynomial resampling filters.
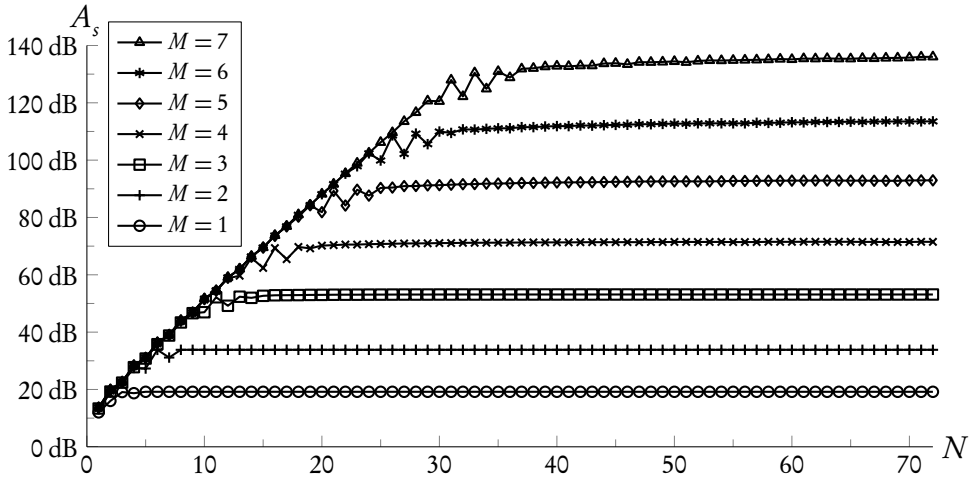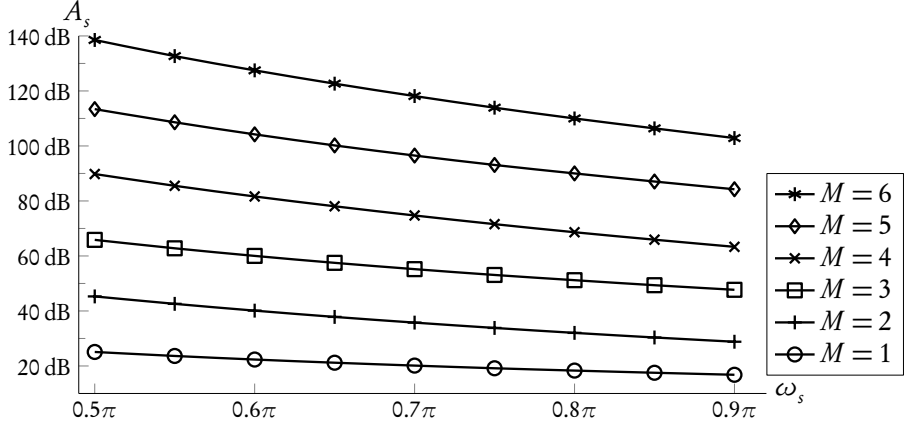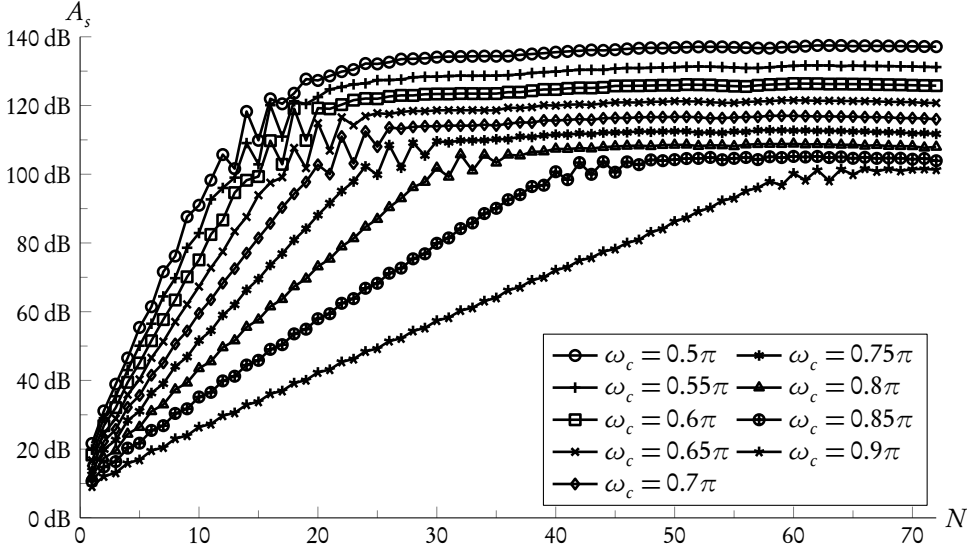
Figure 6.1: Minimum stopband attenuation of the modified Farrow structure as function of M and N. Design specifications: L_{∞} design, $W(X) = 1$, uniform stopband, $\omega_c = 0.75\pi$.

6.3.1 Modified Farrow Structure

6.3.1.1 Role of the Orders M and N

The design parameters for the modified Farrow structure are the subfilter order N and the polynomial order M (3.109). In Figure 6.1, the achievable minimum stopband attenuation A_s (3.228b) is shown for $N = 1, \ldots, 72$ and $M = 1, \ldots, 7$. The coefficient design is performed with respect to the L_{∞} norm using a constant weighting function $W(X) = 1$. A cutoff frequency of $\omega_c = 0.75\pi$ and a uniform stopband specification (3.224) are utilized.

It is observed that for each polynomial order, the trajectory of the stopband attenuation consists of three distinct parts. For small values of N, A_s increases monotonically with an approximately constant slope with respect to the logarithmic scale of this error measure. It is apparent that this slope is very similar for all polynomial orders M. For large values of N, the optimum stopband attenuation is approximately constant and does not increase further. This asymptotic stopband error increases with the order M. A transition region is located between these parts. In this region, the stopband error behaves less deterministic and typically exhibits an oscillation between successive subfilter orders. It is observed that for odd polynomial orders M, the performance increase from an even to an odd N is more pronounced, while A_s increases more slowly or even decreases if N changes from an odd to an even value. Likewise, for M even, the performance improvement is more distinctive if N changes from an odd to an even value than in the converse case. The filter orders N corresponding to this transition region depend on the polynomial order M. In general, the transition region is shifted toward higher values of N as M increases.

From these observations, a guideline for selecting M and N can be derived. First, the polynomial order M should be chosen as the smallest value that meets the required stopband attenuation for large values of N. In general, choosing a larger value for M does not pay off, because it does
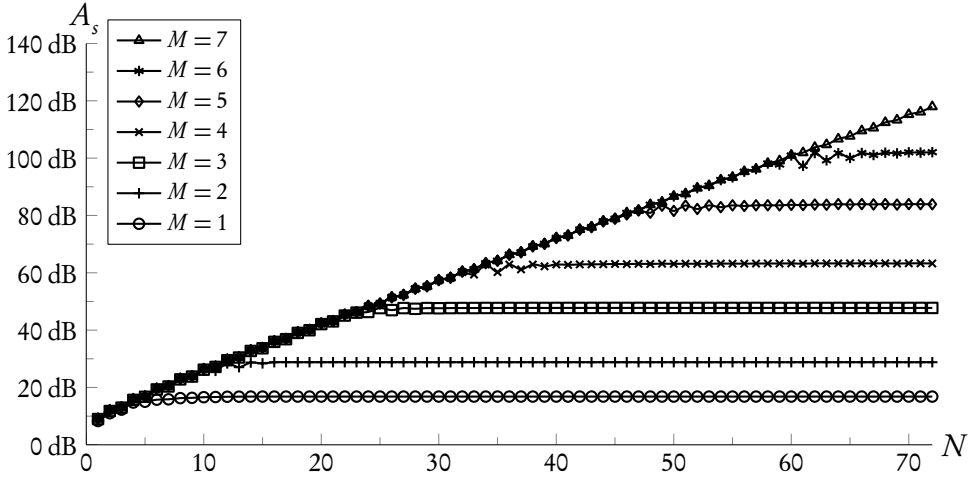
Figure 6.2: Minimum stopband attenuation of the modified Farrow structure as function of M and N. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $\omega_c = 0.9\pi$.

not reduce the required subfilter order N. Second, the order N should be chosen within the transition region or slightly above this interval, because larger values of N no not result in a further improvement of the approximation error. Finally, the parity of N should match that of the polynomial order M to benefit from the oscillations of the stopband error within the transition region.

6.3.1.2 Influence of the Cutoff Frequency

In Figure 6.2, the dependency of the minimum stopband attenuation A_s on M and N is shown for a cutoff frequency $w_c = 0.9\pi$, while the other parameters are identical to those used for Figure 6.1. Apparently, the qualitative behavior is unchanged, but two quantitative changes are observed. First, the asymptotic stopband attenuation for large N is moderately decreased compared to $\omega_c = 0.75\pi$ for all M. Second, the slope of the performance improvement for low values of N is smaller than in the aforementioned case. Consequently, the transition region is shifted to larger values of N if the cutoff frequency is increased.

To further examine the impact of the cutoff frequency ω_c on the stopband attenuation, two investigations have been performed. First, to approximate the asymptotic performance, coefficient sets with very large filter orders ($N = 95$) have been designed in the minimax sense for several cutoff frequencies in the interval $[0.5\pi, 0.9\pi]$. The resulting values of the stopband attenuation are shown in Figure 6.3. This figure shows that the decrease of A_s as a function of ω_c is similar for all polynomial orders M and is relatively moderate.

It is instructive to compare these observations to design rules for FIR filters. An approximate relationship between the passband error δ_p, the stopband error δ_s, the transition band width and the filter order for linear-phase FIR filters designed with respect to the L_∞ norm has been

Figure 6.3: Asymptotic minimum stopband attenuation of the modified Farrow structure as function of M and ω_c. Design specifications: L_∞ design, $W(\omega) = 1$, uniform stopband, $N = 95$.

established in [HRC73, Kai74, CR83]. A simplified formula is given in [Kai74, Sar93]

$$N \approx \frac{-20\log_{10}\left(\sqrt{\delta_p \delta_s}\right) - 13}{\frac{14.6}{2\pi}\Delta\omega}, \tag{6.4}$$

where $\Delta\omega = \omega_s - \omega_p$ denotes the transition band width given as an angular frequency. In minimax filter design, the desired ratio between passband and stopband error is typically enforced by a piecewise constant weighting function with passband and stopband weights $W(X_p)$ and $W(X_s)$ according to (3.229)

$$\frac{W(X_p)}{W(X_s)} = \frac{\delta_s}{\delta_p}.$$

In this case, the stopband error and, equivalently, the minimum stopband attenuation A_s are approximated by functions of N and $\Delta\omega$

$$20\log_{10}\delta_s \approx -\frac{14.6}{2\pi}\Delta\omega N - 20\log_{10}\sqrt{\frac{W(X_s)}{W(X_p)}} - 13 \tag{6.5a}$$

$$A_s \approx \frac{14.6}{2\pi}\Delta\omega N + 20\log_{10}\sqrt{\frac{W(X_s)}{W(X_p)}} + 13. \tag{6.5b}$$

That is, for FIR filters, the minimum stopband attenuation A_s – a logarithmic function of stopband error – is an approximately linear function of both the filter order N and the transition band width $\Delta\omega$. For example, if the width of the transition band is halved, then the minimum stopband attenuation is also approximately halved.
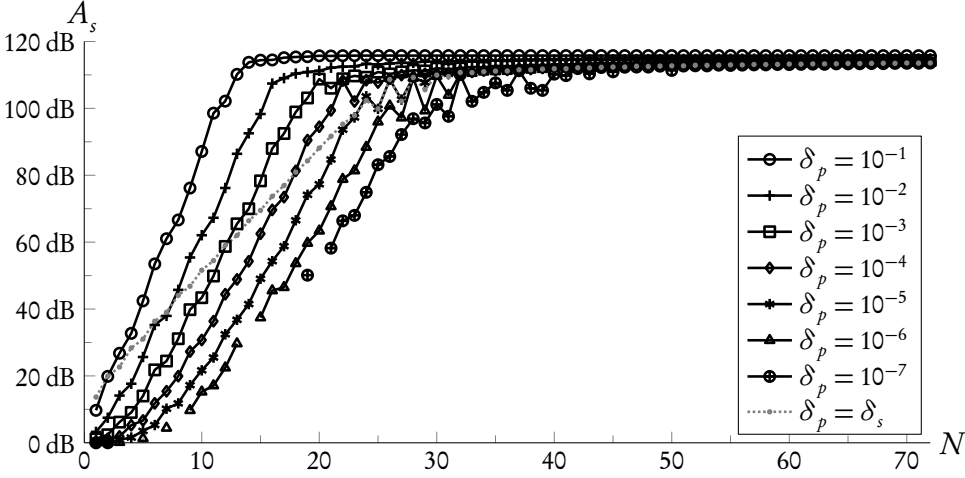
Figure 6.4: Minimum stopband attenuation of the modified Farrow structure as function of N and ω_c. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $M = 6$.

For ASRC systems, the width of the transition band (3.223) is given by $\Delta\omega = 2(\pi - \omega_c)$. As illustrated in Figure 6.3, the decrease of A_s is much lower. For example, the transition band width is halved when the cutoff frequency is increased from 0.5π to 0.75π. In this case, the decrease of the minimum stopband attenuation is about 19 % for $M = 3$ and 18 % for $M = 6$, compared to approximately 50 % expected for a FIR filter design.

The second investigation aims at characterizing the increase of the minimum stopband attenuation A_s for small values of N, that is, in the slope region of the quality trajectory. This slope also determines the location of the transition region. The trajectory of the minimum stopband attenuation A_s is shown in Figure 6.4 for a fixed polynomial order $M = 6$ and several values of ω_c. In Section 6.3.1.1, it has been observed that this slope is nearly independent of the polynomial order M. However, its behavior is easier to analyze for higher values of M, as the slope region is larger.

Apparently, A_s forms an approximately linear function of N in this region irrespective of the cutoff frequency. However, the slope of this function decreases as ω_c increases. To quantify this behavior, a linear regression has been performed to approximate A_s by a linear function

$$A_s(N) = A_s(0) + \Delta A_s N. \tag{6.6}$$

The results of this regression are displayed in Table 6.1. It can be seen that for increasing cutoff frequencies ω_c, corresponding to a decreasing transition band widths $\Delta\omega$, the proportionality factor ΔA_s decreases significantly. At the same time, the offset term $A_s(0)$ exhibits a relatively slow decrease. To analyze these results, a comparison with the design relationships for minimax lowpass FIR filters again proves instructive. Using the approximate formula for the minimum

ω_c	$\Delta\omega$	$M = 6$			$M = 7$		
		$A_s(0)\,[\mathrm{dB}]$	$\Delta A_s\,[\mathrm{dB}]$	$\frac{\Delta A_s}{\Delta\omega}\,[\mathrm{dB}]$	$A_s(0)\,[\mathrm{dB}]$	$\Delta A_s\,[\mathrm{dB}]$	$\frac{\Delta A_s}{\Delta\omega}\,[\mathrm{dB}]$
0.50π	1.0π	16.10	7.59	2.42	16.14	7.66	2.43
0.55π	0.9π	15.78	6.72	2.38	15.47	6.82	2.42
0.60π	0.8π	15.13	5.95	2.37	14.71	6.05	2.41
0.65π	0.7π	14.28	5.22	2.37	14.08	5.27	2.40
0.70π	0.6π	13.26	4.53	2.40	13.67	4.49	2.38
0.75π	0.5π	12.65	3.77	2.40	13.06	3.73	2.38
0.80π	0.4π	12.06	3.02	2.41	12.67	2.97	2.36
0.85π	0.3π	11.37	2.27	2.41	12.05	2.23	2.37
0.90π	0.2π	10.83	1.51	2.41	11.49	1.49	2.36
Sample mean $\overline{x}$				2.40			2.39
Sample standard deviation s				0.0186			0.0238

Table 6.1: Relation between minimum stopband attenuation A_s and cutoff frequency/transition band widths for the modified Farrow structure in the slope region of the quality trajectory.

stopband attenuation of FIR filters (6.5b), the increase ΔA_s is determined by

$$\Delta A_s = A_s(N+1) - A_s(N)$$
$$\approx \frac{14.6}{2\pi}\Delta\omega \tag{6.7a}$$
$$\approx 2.34\Delta\omega\,, \tag{6.7b}$$

that is, $A_s(N)$ is an approximately linear with respect to N for FIR filters, which is consistent with the conjectured behavior for the modified Farrow structure (6.6). Moreover, ΔA_s in (6.7a) is proportional to the transition band width $\Delta\omega$ with the proportionality constant given in (6.7b). To compare this relationship to the behavior of the modified Farrow structure, the values ΔA_s for different cutoff frequencies in Table 6.1 are divided by the respective transition band width. The results are shown in the column "$\Delta A_s/\Delta\omega$". It is observed that this ratio is very similar for all values of ω_c. Its sample mean and sample standard deviation are given by $\overline{x} = 2.396$ and $s = 0.0197$, respectively.

From this observation, it appears that in the slope region of the performance trajectory, the achievable stopband attenuation of the modified Farrow structure resembles the behavior of a linear-phase lowpass filter relatively closely. In particular, the increase of the stopband attenuation ΔA_s is approximately proportional to the transition band width $\Delta\omega$ in both cases. Although the mean of the proportionality constant $\overline{x} = 2.396$ comes relatively close to the constant value 2.34 derived for FIR filters (6.7b), this seeming coincidence should not be overrated in the author's opinion. Considering the conceptual differences between the continuous-time frequency response of the Farrow structure and the discrete-time frequency response of a FIR filter, more research is needed to quantify the interrelations between these filters.

This investigation has been repeated for polynomial order $M = 7$, and the results are shown

Figure 6.5: Minimum stopband attenuation of the modified Farrow structure as function of the passband error δ_p. Design specifications: L_∞ design, $W(X) = 1$, uniform stopbands, $\omega_c = 0.75\pi$, $M = 6$.

in the right part of Table 6.1. It can be observed that the values for the proportionality factor (sample mean $\overline{x} = 2.389$, sample standard deviation $s = 0.0238$) are very similar to the results for $M = 6$.

6.3.1.3 Fixed Passband Error Specifications

In the previous investigations, the design error was weighted uniformly. However, in many applications, it is sensible to use distinct specifications for the passband and stopband error. For instance, in audio applications it is often advantageous to maximize the stopband attenuation for a given maximum passband ripple.

In Figure 6.5, the achievable stopband error is shown as a function of the subfilter order for several choices of a fixed passband error limit δ_p. A design with uniform error weighting $\delta_p = \delta_s$ as described above is included for comparison. Missing points in the quality function, especially for tight passband error limits, represent designs that failed to achieve the required error limit due to infeasibility of the corresponding optimization problem.

It is observed that the asymptotic stopband attenuation for large N is similar for all choices of δ_p. Likewise, the slopes in the region of approximately linear increase of A_s are of comparable magnitude. However, the offset where this slope region starts is shifted toward higher values of N as the admissible passband error decreases. In contrast, the slope for the uniform error weighting $\delta_p = \delta_s$ is significantly lower than for the designs using fixed passband errors.

Again, it is worthwhile to compare this behavior to the design relationships for FIR filters. According to (6.4), the passband and stopband error show an approximately inverse relation for minimax FIR filter design. Solving this approximate relation for a fixed passband error δ_p

results in

$$A_s \approx 2\frac{14.6}{2\pi}\Delta\omega + 20\log_{10}\delta_p + 26. \tag{6.8}$$

That is, for FIR filters, a fixed passband error limit δ_p results in a constant offset for the minimum stopband attenuation. In contrast, the proportionality factor determining the effect of the filter order N on A_s is independent of this limit. However, in comparison to a design based on a weighted error specification (6.5b), this proportionality factor is twice as large.

Applied to the modified Farrow structure, it becomes apparent that this approximate design relation matches the qualitative behavior in the slope region of the minimum stopband attenuation relatively well. However, for decreasing passband error limits, the slope of the quality trajectory decreases moderately, which is not modeled by the expression (6.8).

Notwithstanding this quantitative deviation, this analogy to the behavior of linear-phase FIR filters clarifies the role of passband error specifications in the design of the modified Farrow structure. The qualitative similarity in the slope region of the quality trajectory suggests that the adherence to a passband error specification is connected to the properties of the subfilters $C_m(z)$, which are linear-phase filters in case of the modified Farrow structure In contrast, the upper limit for the achievable minimum stopband attenuation (or, equivalently, the bound for the maximum stopband error δ_s) is virtually independent of a passband error specification. This emphasizes the preeminent role of the stopband error metric for this class of ASRC filters.

Summarizing, these observations show that fixed passband error limits can be readily incorporated into design methods for the modified Farrow structure. Depending on other design parameters, such design specifications may enable to meet the requirements of an application with reduced computational complexity. However, the achievable improvements are tightly limited in most cases. For instance, it is unlikely that the polynomial order M, which is the most influential parameter for the computational effort, can be reduced due to a relaxed passband error specification.

6.3.1.4 Nonuniform Stopband Specifications

Both uniform and nonuniform stopband specifications, as defined in Section 3.8.1, are used in ASRC design. While nonuniform stopbands require the input signal to be bandlimited to the cutoff frequency ω_c, the use of multiple transition regions offers additional degrees of freedom that potentially result in an improved performance. In Figure 6.6, the achievable minimum stopband attenuation is shown as a function of M and N for a L_∞ design with uniform error weighting. It is observed that the general behavior of both designs is relatively similar. In particular, the dependency on the subfilter order in the slope region is nearly identical. Likewise, the asymptotic stopband attenuation for large values of N appears to be independent of the stopband specification. However, the transition between these regions is different. As described above, designs with respect to a uniform stopband specification show a pronounced transition region with fluctuations between successive subfilter orders as well as a region where the stopband error converges to its asymptotic limit. In contrast, the transition between the linear increase and the region determined by the upper limit of A_s is nearly immediate for nonuniform stopband designs.

To illustrate the differences between both specifications, design examples for $M = 6$, $N = 40$ are shown in Figure 6.7. It is seen that the continuous frequency responses are markedly different.
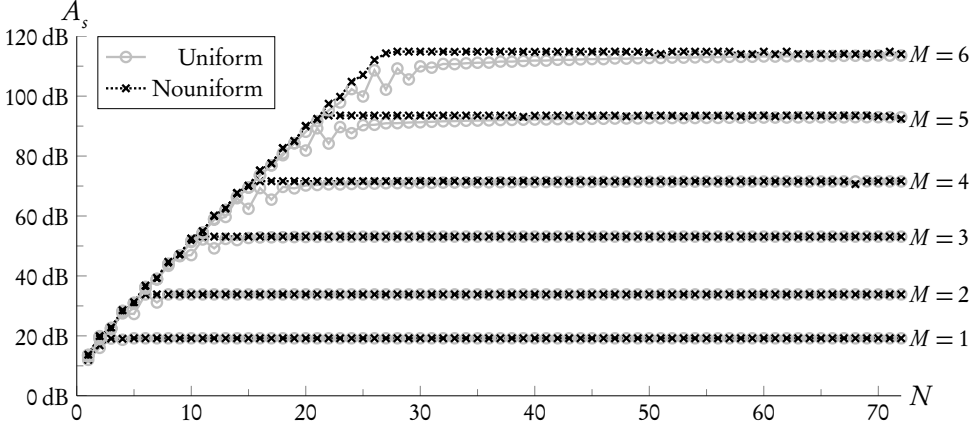
Figure 6.6: Minimum stopband attenuation of the modified Farrow structure for uniform and nonuniform stopbands. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $\omega_c = 0.75\pi$.

While the design with uniform stopband region exhibits an approximately equiripple error over wide regions of X_s, the stopband error resulting from the nonuniform design specification is typically lower, but exhibits a significant magnitude in the don't care regions of the design specification. However, because the maximum error of the nonuniform stopband design is determined by the frequency response error at the edges of the multiple stopband regions, the achieved minimum stopband attenuation is relatively similar for both design specifications (114.8 dB for nonuniform stopbands and 111.8 dB for uniform stopbands).

In summary, the potential performance gains from using nonuniform stopbands are tightly limited in most cases, and the same stopband attenuation is typically achieved by a uniform stopband design, possibly at the expense of a slightly increased subfilter order. It is therefore questionable for most applications whether the small potential performance gains pay off compared to the risks due to insufficient attenuation in the transition bands.

6.3.1.5 Weighted Least Squares Designs

So far, the performance evaluations have been performed with respect to the L_∞ norm. To assess the influence of the error norm used in the design process, the evaluation of the design error is repeated for the L_2 norm. In Figure 6.8, the L_2 error δ_2, which is the square root of the weighted squared error E_2 (3.226a)

$$\delta_2 = \sqrt{E_2} = \sqrt{\int\limits_X \left(W(\omega) \left| H_c(\mathrm{j}\omega) - \widehat{H}_c(\mathrm{j}\omega) \right| \right)^2 \mathrm{d}\omega}, \qquad (6.9)$$

is displayed as a function of M and N for designs with respect to the L_2 norm. The optimization is performed for an uniform stopband specification, a constant weighting function $W(X) = 1$, and a cutoff frequency $\omega_c = 0.75\pi$. The error δ_2 is obtained by numerical integration with an upper frequency limit $\omega_{up} = 32\pi$.
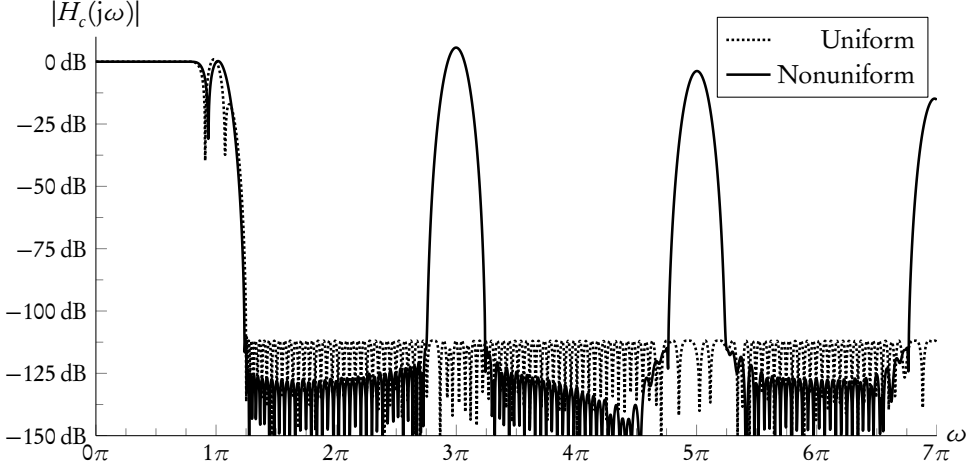
Figure 6.7: Design example for the modified Farrow structure with uniform and nonuniform stopbands. Design specifications: L_∞ design, uniform error weighting $W(X) = 1$, $\omega_c = 0.75\pi$, $M = 6$, $N = 40$.

Apart from the different semantics of the error metrics and the inverted orientation of δ_2 compared to the stopband attenuation A_s, it is observed that the qualitative behavior of the L_2 error is very similar that of the L_∞ designs shown in Figure 6.1. The distinct regions of the error trajectory reported in Section 6.3.1.1 are clearly found in the behavior of the L_2 error. Specifically, the region of approximately linear decrease of the error (measured on a logarithmic scale), the lower error limit for large subfilter orders N depending on the polynomial order M, and the form of the transition regions closely resemble their counterparts in the L_∞ design experiments.

These observations suggest that the performance characteristics of the modified Farrow structure investigated in this section are not specific either to L_∞ designs or to particular error measures such as the minimum stopband attenuation. In contrast, they very likely reflect a general property of this class of ASRC systems. In particular, the rather strict dependency of achievable design error on the polynomial order M is not characteristic to the L_∞ norm, but holds for designs with respect to other error measures as well.

6.3.1.6 Metrics for the Computational Effort

The modified Farrow structure consists of two distinct parts: the bank of subfilters $C_m(z)$ and the Horner scheme for polynomial evaluation. This partitioning is also reflected in the performance metrics for this structure. While the subfilters $C_m(z)$ operate on the sequence of input samples, the Horner scheme is evaluated for each output sample. Consequently, the addition and multiplication rates follow from the instruction counts stated in Section 3.6.2.

$$R^+ = \begin{cases} (M+1)Nf_i + Mf_o, & N \text{ odd} \\ \left[(M+1)(N-1) + \left\lceil \frac{M+1}{2} \right\rceil \right] f_i + Mf_o, & N \text{ even} \end{cases} \qquad (6.10a)$$
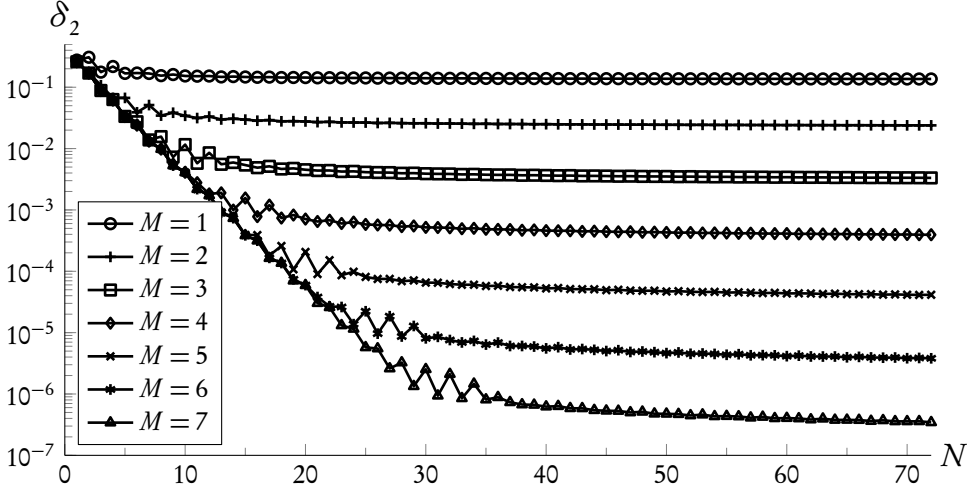
Figure 6.8: L_2 error of the modified Farrow structure as function of M and N. Design specifications: L_2 design, $W(X) = 1$, uniform stopband, $\omega_c = 0.75\pi$.

$$R^* = \begin{cases} (M+1)\frac{N+1}{2}f_i + Mf_o, & \text{odd} \\ \left[(M+1)\frac{N}{2} + \left\lceil \frac{M+1}{2} \right\rceil\right]f_i + Mf_o, & N \text{ even} \end{cases}. \tag{6.10b}$$

The output values of the subfilters $C_m(z)$ form the results of the processing of input samples. Consequently, the data memory requirements amount to $M+1$ values per input sample. Likewise, the memory bandwidth required for computing one output samples is $M+1$, because a complete set of subfilter outputs is required to evaluate the polynomial of order M.

The implementation delay basically results from the delay required to implement the subfilters $C_m(z)$ in causal form, which is $N/2$ samples. However, this value is also affected by the range of the intersample position variable μ. Thus, the implementation delay follows from the causality requirements derived in 3.5.5 and amounts to

$$D_{impl} = \frac{N-1}{2}.$$

6.3.1.7 Conclusions

In this section, the effect of different parameters and design specifications on the achievable resampling quality of the modified Farrow structure has been investigated. The preeminent roles of the polynomial order M and the stopband error forms the most important observation. The polynomial order M effectively establishes a lower limit on the stopband error that cannot be improved significantly if the subfilter order N is further increased or if other design specifications are relaxed. In contrast, the effects of other design specifications, such as the cutoff frequency, passband error constraints or the form of the stopband region, can be controlled to a large extent by an appropriate choice for the subfilter order N.
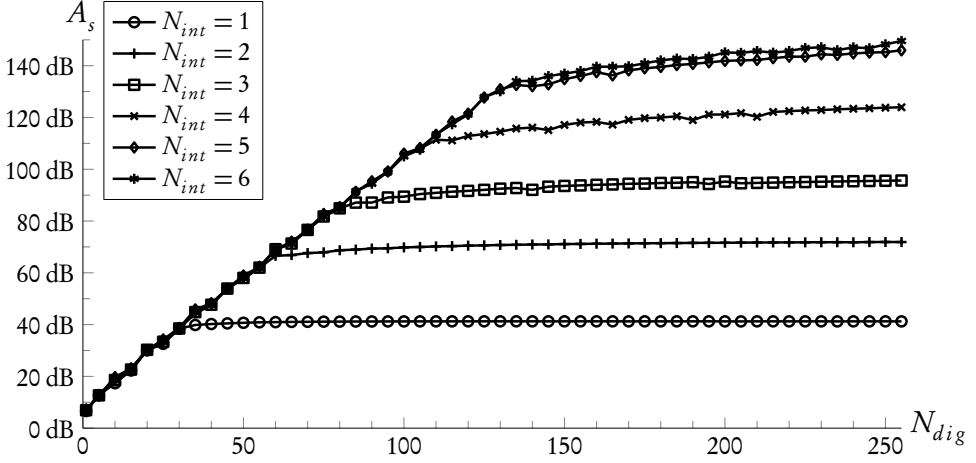
Figure 6.9: Minimum stopband attenuation of ASRC structures using oversampling as function of N_{dig} and N_{int}. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $H_{int}(j\omega)$ according to OIB design, $\omega_c = 0.75\pi$, $L = 4$.

The distinctive role of the polynomial order also affects the performance metrics for the modified Farrow structure significantly. The number of instructions is strongly influenced by the number of subfilters and the order of the polynomial to be evaluated, which are both determined by M. Moreover, both the storage requirements per input sample and the memory bandwidth per computed output sample depend directly on the polynomial order.

6.3.2 Structures Incorporating Integer-Ratio Oversampling

The main design parameters for ASRC structures based on oversampling are the oversampling ratio L, the order N_{dig} of the discrete-time prefilter and the characteristics of the continuous-time resampling filter $H_{int}(j\omega)$. For the reasons stated in Section 5.5, only symmetric piecewise polynomial resampling functions with polynomial order $M = N_{int}$ are considered here. This class of resampling filters includes widely-used functions such as Lagrange interpolators, spline basis functions or MOMS interpolation kernels. In most design experiments, the optimized image band attenuation (OIB) design proposed in Section 5.5.3 is utilized, because it enables the best resampling performance of all fixed resampling functions in the majority of designs.

For performance evaluation, only discrete-time prefilters $H_{dig}(e^{j\omega})$ designed with the overall optimization scheme introduced in Section 5.4 are considered. This choice is motivated by the ability to provide optimal designs with respect to a flexible set of specifications.

6.3.2.1 Role of the Orders N_{dig} and N_{int} and the Oversampling Ratio L

In Figure 6.9, the minimum stopband attenuation A_s for a minimax design with uniform error weighting is shown as a function of the filter orders N_{dig} and N_{int} for a fixed oversampling ratio $L = 4$ and cutoff frequency $\omega_c = 0.75\pi$. For this investigation, $H_{int}(j\omega)$ is designed with respect
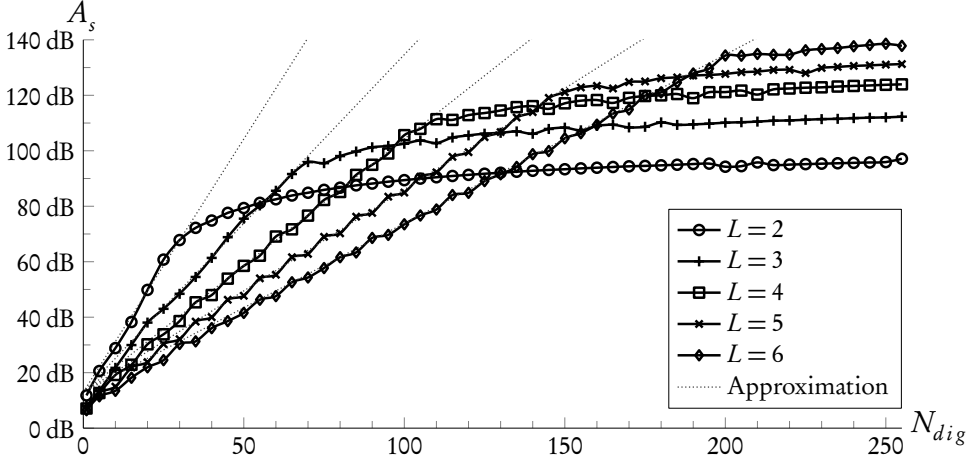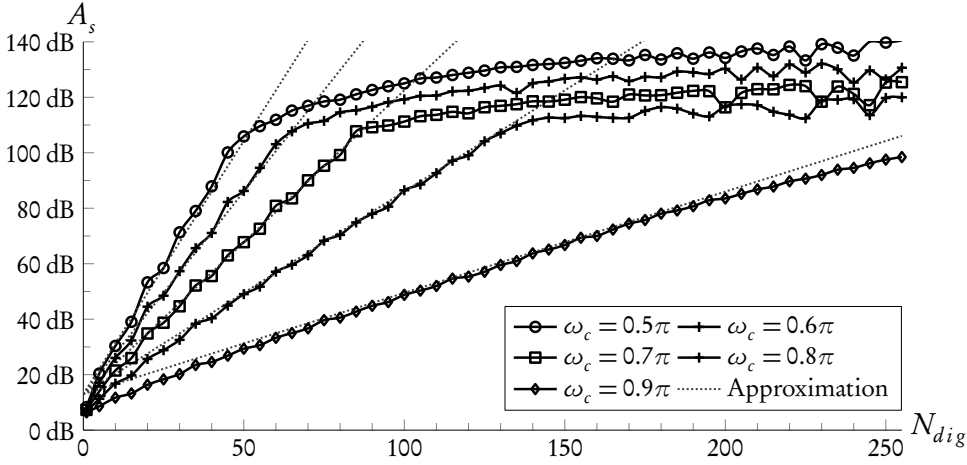
Figure 6.10: Minimum stopband attenuation of ASRC structures using oversampling as function
of N_{dig} and L. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband,
$H_{int}(j\omega)$ according to OIB design, $\omega_c = 0.75\pi$, $N_{int} = 4$.

to the optimized image band (OIB) criterion introduced in Section 5.5.3. However, as will be
shown later in Section 6.3.2.3, the qualitative behavior for other classes of resampling filters is
identical.

It is observed that the trajectory of A_s consists of several distinct regions for each N_{int},
resembling the behavior of the modified Farrow structure observed, for instance, in Figure 6.1.
For low prefilter orders, the increase of the stopband attenuation is approximately linear (on the
logarithmic scale underlying A_s) and independent of the order N_{int}. For higher values of N_{dig},
the achievable stopband attenuation increases only slowly and apparently approaches an upper
limit.

This qualitative behavior is in accordance with the roles of the two components $H_{dig}(e^{j\omega})$
and $H_{int}(j\omega)$ characterized in Section 5.4. For low prefilter orders, the quality is limited by
image suppression capabilities of $H_{dig}(e^{j\omega})$. In the asymptotic limit, that is, for large values of
N_{dig}, the image attenuation of $H_{int}(j\omega)$ in the passband image regions of $H_{dig}(e^{j\omega})$ becomes
the limiting factor. In between, if the image attenuation capability of $H_{dig}(e^{j\omega})$ is sufficiently
large, the overall performance is governed by the adaptation of the discrete-time prefilter to
the frequency response $H_{int}(j\omega)$. Specifically, this comprises the compensation of passband
errors and the shaping of the transition regions of $H_{dig}(e^{j\omega})$. For this reason, the increase of the
achievable quality in this intermediate region is relatively slow.

It is observed that for each value of N_{int}, there is an effective upper limit on the achievable
stopband attenuation. The quality improvement for successive values of N_{int} becomes smaller as
this order increases.

The achievable stopband attenuation as a function of the upsampling ratio L and the prefilter
order N_{dig} is shown in Figure 6.10 for a fixed order $N_{int} = 4$ of $H_{int}(j\omega)$. As in the previous
design example, the performance trajectory consists of three distinct regions. For high prefilter

orders N_{dig}, the achievable stopband attenuation is nearly constant. This asymptotic error limit increases with the oversampling ratio L, but the improvement between successive values of L diminishes. Before this asymptotic limit is reached, the performance trajectory exhibits a transition region as characterized in the previous design experiment. For relatively low prefilter orders, the increase of A_s is approximately linear. The slope within this region depends on the oversampling ratio L and decreases as this ratio increases. Consequently, the transition region is shifted towards higher values of N_{dig} for higher oversampling ratios.

The different slopes for low values N_{dig} are explainable from the characteristics of the prefilter $H_{dig}(e^{j\omega})$. As in rational SRC, the primary role of this filter is to provide lowpass filtering with a cutoff frequency of ω_c/L and a transition band in the interval $(\omega_c/L, (2\pi - \omega_c)/L)$ (see e.g. [CR83]). That is, the transition band width is

$$\Delta\omega = \frac{2(\pi - \omega_c)}{L}. \tag{6.11}$$

According to equation (6.5b), the minimum stopband attenuation for a linear-phase lowpass filter is an approximately linear function with a slope proportional to $\Delta\omega$. Applying the uniform weighting function $W(X_p) = W(X_s) = 1$ used in the design experiments results in the approximation

$$A_s \approx 14.6\frac{\pi - \omega_c}{\pi L} + 13. \tag{6.12}$$

In Figure 6.10, it is observed that this approximation matches the actual stopband attenuation in the slope region of the performance trajectories well.

6.3.2.2 Influence of the Cutoff Frequency

The dependency between the cutoff frequency ω_c and the minimum stopband attenuation is investigated in Figure 6.11 for a fixed upsampling ratio $L = 4$ and resampling order $N_{int} = 4$. It can be seen that the slope in the region of linear increase decreases as the cutoff frequency attains higher values. As in case of varying upsampling ratios, this behavior is well matched by the approximation (6.12).

It is observed that the maximum achievable stopband attenuation decreases rather slowly if the cutoff frequency is increased. This behavior is similar to the behavior of the modified Farrow structure described in Section 6.3.1.2. However, the prefilter order N_{dig} to reach such attenuation values increases considerably.

6.3.2.3 Influence of the Continuous-Time Resampling Function

The effect of the characteristics of the continuous-time resampling filter $H_{int}(j\omega)$ is examined for fixed parameter settings $L = 4$, $N_{int} = 4$ and $\omega_c = 0.75\pi$. The results are illustrated in Figure 6.12. For low prefilter orders, that is, in the region of linearly increasing A_s, the stopband attenuation is virtually independent of the characteristics of $H_{int}(j\omega)$. In contrast, the point of transition where the trajectory departs from the region of approximately linear increase depends on the properties of the continuous-time resampling function. In general, the sophisticated designs proposed in this work enable significantly higher stopband attenuations for a given order N_{int} than, for instance, conventional Lagrange interpolators.

Figure 6.11: Minimum stopband attenuation of ASRC structures using oversampling as function of N_{dig} and cutoff frequency ω_c. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $H_{int}(j\omega)$ according to OIB design, $L = 4$, $N_{int} = 4$.

6.3.2.4 Fixed Passband Error Specifications

The influence of different ratios between passband and stopband error is depicted in Figure 6.13, where the minimum stopband attenuation is shown for a set of fixed passband error specifications.

The perceived effect is similar to that of the modified Farrow structure described in Section 6.3.1.3. The asymptotic performance for large values of N_{dig} shows a relatively slow decrease for increasingly strict passband specifications. In the region of approximately linear increasing A_s, the slope of the functions is relatively similar, while the fixed passband error causes an offset to this function. Applying the transition band width (6.11) to the design relation for FIR filters (6.8) results in the approximation

$$A_s \approx \frac{29.2(\pi - \omega_c)}{\pi L} + 20 \log_{10} \delta_p + 26 \,. \tag{6.13}$$

In Figure 6.13, it is seen that these linear approximations match the actual performance trajectories relatively well. However, as in case of the modified Farrow structure, the actual slope decreases as δ_p attains very small values.

According to (6.8), the slope of A_s for a fixed passband error is twice as large as for a design with a weighted error specification. To check this hypothesis, the actual performance trajectory for a design with uniform weighting is included in Figure 6.13, where it is denoted by $\delta_p = \delta_s$. Apparently, its slope is significantly lower than for the designs with fixed passband error limits. Moreover, it matches the linear approximation (6.12) relatively well.
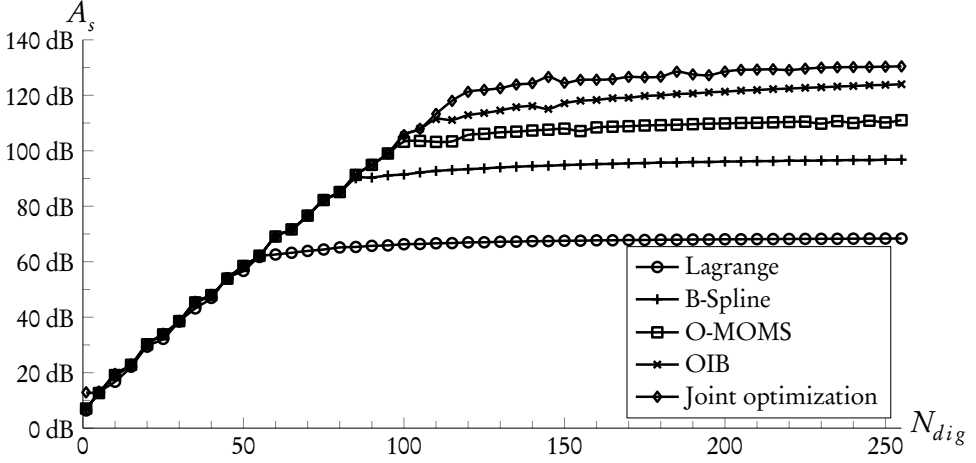
Figure 6.12: Minimum stopband attenuation of ASRC structures using oversampling for different resampling filters $H_{int}(j\omega)$. Design specifications: L_∞ design, $W(X) = 1$, uniform stopband, $\omega_c = 0.75\pi$, $L = 4$, $N_{int} = 4$.

6.3.2.5 Nonuniform Stopband Specifications

In Figure 6.14, the effect of uniform and nonuniform stopband specifications on the achievable stopband attenuation are shown for an example design with $L = 4$, $\omega_c = 0.75\pi$ and uniform error weighting. As in most previous design experiments, an OIB design objective is used for the continuous-time resampling function $H_{int}(j\omega)$. It can be observed that for nonuniform stopbands, the minimum stopband attenuation is slightly lower than that of a comparable design with a uniform stopband, but the differences diminish relatively quickly in most cases as the prefilter order N_{dig} increases.

Moreover, it can be seen that for nonuniform stopbands, the transition between the region of linear increase and the flat behavior for large values of N_{dig} is almost immediate. This behavior is readily explained by the role of the transition band of $H_{dig}(e^{j\omega})$ described earlier in this section. Basically, the performance of both designs differs for intermediate values of N_{dig}, that is, between the region of approximately linear performance increase and the constant error bound for large prefilter orders.

For uniform stopbands, the achievable attenuation in this region is determined by the adaptation of $H_{dig}(e^{j\omega})$ in the transition band image regions of this filter. In contrast, these image bands are excluded in a nonuniform stopband specification. Thus, the performance trajectories change abruptly from the region where the performance is dominated by the stopband rejection of the discrete-time prefilter to the asymptotic error imposed by the minimum image attenuation of $H_{int}(j\omega)$.

Consequently, for nonuniform stopband specifications, the primary objective of the proposed overall optimization scheme for $H_{dig}(e^{j\omega})$ is to compensate passband deviations caused by $H_{int}(j\omega)$. Thus, the achievable improvements of this design method are more limited than in case of uniform stopbands.
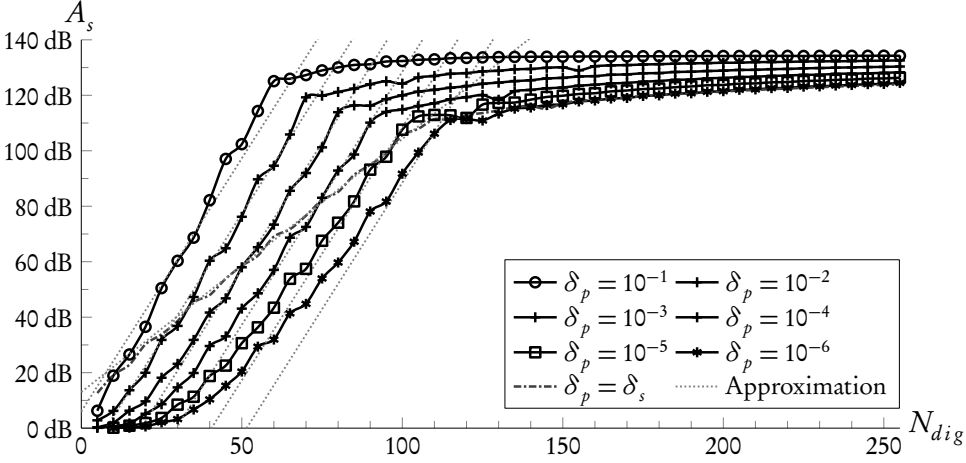
Figure 6.13: Minimum stopband attenuation of ASRC structures using oversampling as function of the passband error δ_p. Design specifications: L_∞ design, uniform stopband, $\omega_c = 0.75\pi$, $L = 4$, $N_{int} = 4$.

6.3.2.6 Weighted Least Squares Designs

In Figure 6.15, the weighted least squares error δ_2 is depicted as a function of the orders M and N. The designs have been performed with respect to the L_2 norm with constant error weighting $W(X) = 1$, a uniform stopband specification and cutoff frequency $\omega_c = 0.75\pi$. As in case of the modified Farrow structure, the qualitative behavior of the approximation error is very similar to the L_∞ error for minimax designs considered in Section 6.3.1.5. In particular, the region of linearly decreasing error (with respect to a logarithmic scale) as well as the performance limitation depending on the resampling filter order N_{int} are clearly represented in the error functions. Moreover, the prefilter orders where the performance trajectories change from a uniform increase to an approximately constant behavior is relatively similar for L_∞ and L_2 designs. This observation supports the conjecture that, as in case of the modified Farrow structure, the qualitative behavior of the performance is intrinsic to the implementation structure. That is, this behavior is not specific to a particular design method or error norm.

For $N_{int} = 6$, the error function contains some values where the design error increases non-systematically. Very likely, these outliers are caused by numerical ill-conditioning in the least-squares optimization. It is noted that the absolute design error of the optimization problems in question is very small, so small perturbations in the numerical optimization are likely to result in considerable relative errors. Moreover, ill-conditioning for large problem sizes is a known problem of WLS filter design with transition bands, e.g. [BBS94, LVKL96, LD99].

6.3.2.7 Metrics for the Computational Effort

The computational costs reflect the two components of ASRC systems based on oversampling and continuous-time resampling functions. The computation for the discrete-time sample rate conversion process is performed for each input sample. As integer-ratio SRC is efficiently
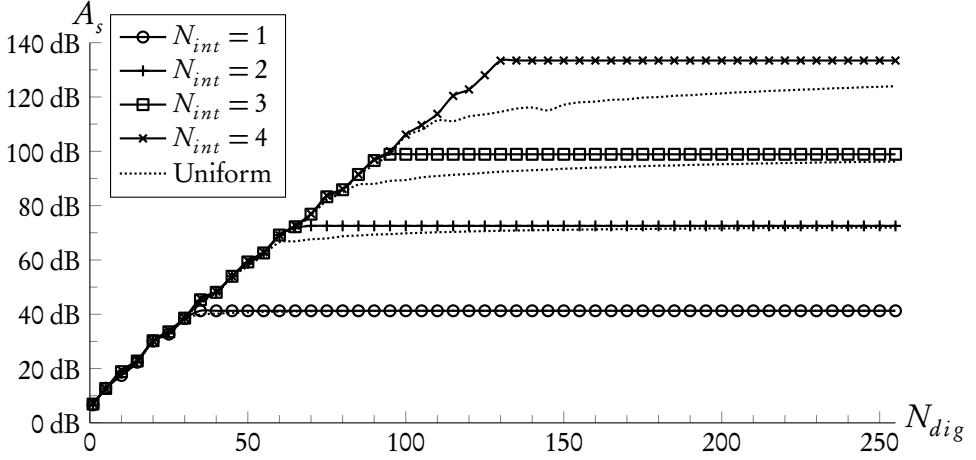
Figure 6.14: Minimum stopband attenuation of ASRC structures using oversampling for uniform and nonuniform stopbands. Design specifications: L_∞ design, $W(X) = 1$, $H_{int}(j\omega)$ according to OIB design, $\omega_c = 0.75\pi$, $L = 4$.

performed by polyphase filtering (see Section 3.4.2 or [BBC76, CR81, CR83]), this amounts to the evaluation of L FIR filters of length $(N_{dig} + 1)/L$. It is assumed that the discrete-time prefilter $H_{dig}(e^{j\omega})$ is zero-padded such that the length $N_{dig} + 1$ is an integral multiple of L. As the polyphase components do not exhibit coefficient symmetries in general, the numbers of instructions for each input sample are given by

$$N_i^+ = N_{dig} + 1 - L \tag{6.14a}$$

$$N_i^* = N_{dig} + 1 \tag{6.14b}$$

$$N_i = 2N_{dig} + 2 - L. \tag{6.14c}$$

The number of operations per output sample depends on the structure of the continuous-time resampling function $H_{int}(j\omega)$. In general, the functions considered in this work can be implemented as modified Farrow structures of order $M = N = N_{int}$. In this case, the instruction counts follow from the complexity figures of Section 3.6.2

$$N_o^+ = \begin{cases} N_{int}^2 + \frac{3}{2}N_{int}, & N_{int} \text{ even} \\ N_{int}^2 + 2N_{int}, & N_{int} \text{ odd} \end{cases} \tag{6.15a}$$

$$N_o^* = \begin{cases} \frac{1}{2}N_{int}^2 + 2N_{int} + 1, & N_{int} \text{ even} \\ \frac{1}{2}N_{int}^2 + 2N_{int} + \frac{1}{2}, & N_{int} \text{ odd} \end{cases} \tag{6.15b}$$

$$N_o = \begin{cases} \frac{3}{2}N_{int}^2 + \frac{7}{2}N_{int} + 1, & N_{int} \text{ even} \\ \frac{3}{2}N_{int}^2 + 4N_{int} + \frac{1}{2}, & N_{int} \text{ odd}. \end{cases} \tag{6.15c}$$

However, if Lagrange interpolation is utilized, more efficient implementation structures than the modified Farrow structure exist as argued in Section 3.6.3. Using the advantageous linear-
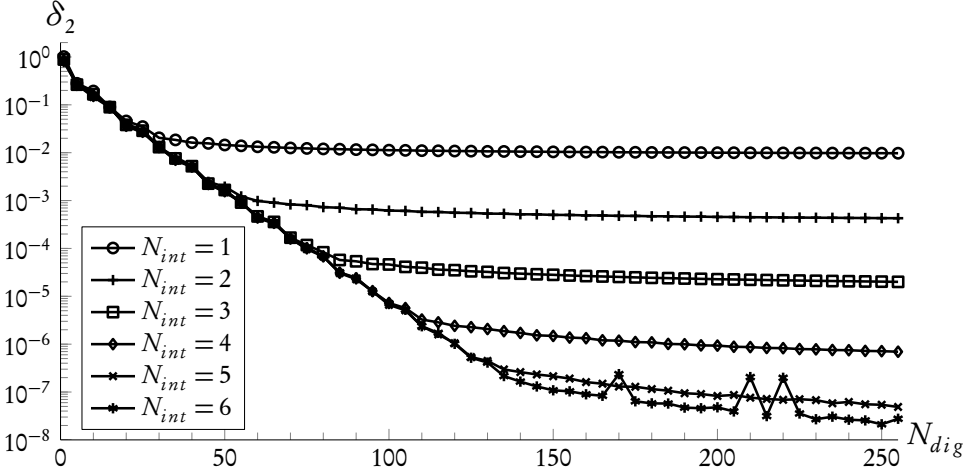
Figure 6.15: L_2 error of ASRC structures using oversampling as function of N_{dig} and N_{int}.
Design specifications: L_2 design, $W(X) = 1$, uniform stopband, $H_{int}(j\omega)$ according
to OIB design, $\omega_c = 0.75\pi$, $L = 4$.

complexity structure proposed in Section 3.6.3.11, the instruction counts amount to

$$N_o^+ = 2N_{int} - 1 \tag{6.16a}$$

$$N_o^* = 3N_{int} + \left\lceil \frac{N_{int} + 1}{2} \right\rceil + 1 \tag{6.16b}$$

$$N_o = 6N_{int} + \left\lceil \frac{N_{int} + 1}{2} \right\rceil . \tag{6.16c}$$

The storage required for each input sample amounts to L data values that are generated by
the integer-ratio SRC process. The memory access bandwidth is determined by the number of
values accessed by the continuous-time resampling kernel. Thus, it amounts to $N_{int} + 1$ values
per output sample.

The implementation latency of this algorithm is the sum of the implementation delays of the
components $H_{dig}(e^{j\omega})$ and $H_{int}(j\omega)$. A linear-phase prefilter $H_{dig}(e^{j\omega})$, introduces a delay of
$N_{dig}/2$ samples. Because the resampling filters $H_{int}(j\omega)$ considered here are symmetric piecewise
polynomial filters, the implementation delay of this component amounts to $(N_{int} - 1)/2$ samples
as explained in Section 6.3.1.6. However, as both components conceptually operate on a L-fold
increased sampling frequency, the corresponding values are scaled by $1/L$. In combination, the
total implementation latency of this structure is given by

$$D_{impl} = \frac{N_{dig} + N_{int} - 1}{2L} . \tag{6.17}$$

6.3.2.8 Conclusions

The analyses performed in this section show that the different design parameters and objectives have distinctive effects on the achievable quality of systems incorporating integer-ratio oversampling. First of all, as for the modified Farrow structure, the maximum stopband error δ_s, or equivalently the minimum stopband attenuation A_s, is identified as the most important design objective. The design objective for the continuous-time resampling function $H_{int}(j\omega)$ and the order of this component as well as the oversampling ratio L determine the achievable quality of the structure. The choice of the order N_{dig} of the discrete-time prefilter depends on these variables. Below a specific limit determined by N_{int} and L as well as the cutoff frequency ω_c and optionally fixed passband error constraints, the stopband attenuation is an approximately linear function of N_{dig}. Above this limit, a further increase of the prefilter order does not improve the design quality significantly. Apart from this dependency, the influence of the cutoff frequency, fixed passband error requirements or nonuniform stopband specifications on the maximally achievable design quality are relatively mild. Conversely, the effects of these objectives on the design quality can be controlled by the order of the discrete-time prefilter for the most part.

6.4 Trade-offs between Quality and Computational Effort

In the preceding section, the impact of a multitude of design parameters and objectives on the achievable quality of the different implementation structures for ASRC has been investigated. However, the selection of an algorithm that is optimally suited for an application and a given target platform remains a difficult task that offers numerous degrees of freedom. To aid in this process, the computational resources required by the different implementation structures are examined and compared in this section.

The basic idea underlying this analysis is to compare the performance metrics defined in Section 6.2.2 for different implementation structures as a function of a design objective. In the preceding section, it became apparent that the stopband error, represented by the minimum stopband attenuation A_s, has a particular importance compared to other design specifications. Moreover, the required stopband attenuation typically has a significant impact on the computational complexity of an ASRC method. For these reasons, A_s is used as the primary design objective in the trade-off analyses.

The general procedure for the analysis is as follows: Coefficient designs are performed for an extensive set of combinations of design parameters, and the quality of the obtained ASRC filters is evaluated. For the modified Farrow structure, the design paremeters M and N are varied in the ranges $M = 1, 2, \ldots, 7$ and $N = 1, 2 \ldots, 72$. Likewise, for structures incorporating integer-ratio oversampling, the respective parameter ranges are $L = 2, 3, \ldots, 6$, $N_{int} = 1, 2, \ldots, 7$ and $N_{dig} = 5, 10, \ldots, 255$.

A set of design objectives for the required minimum stopband attenuation is formed as an equidistant grid $A_s = 40, 45, \ldots, 125\,\text{dB}$. For each specification, the ASRC designs that fulfill this objective are selected, and the metrics for the computational effort for these designs are evaluated. Among these performance numbers, the minimum value denotes the minimum computational cost to fulfill the prescribed performance goal. It forms a single value in the trade-off function between quality and computational cost.

Because an exhaustive investigation is infeasible, several design specifications are set to fixed
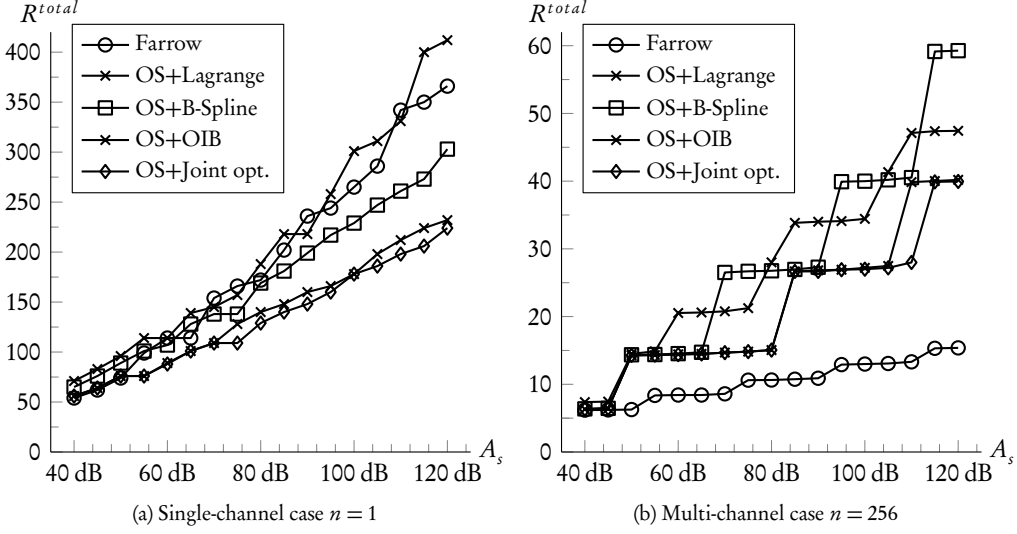
Figure 6.16: Minimal arithmetic complexity as a function of the minimum stopband attenuation for different ASRC implementation structures.

values. In particular, the design is performed with respect to the L_∞ norm with uniform passband and stopband error weightings, a uniform stopband specification and a cutoff frequency $\omega_c = 0.75\pi$. Strictly speaking, the trade-off functions are applicable only to these parameter settings. However, it is very probable that the qualitative behavior of the cost functions is similar for other design specifications. This expectation is supported by the analyses performed in Section 6.3.

6.4.1 Arithmetic Complexity

As stated in Section 6.2.2, the arithmetic complexity of an ASRC algorithm is influenced by the conversion ratio R. For multichannel applications such as WFS, the effort also depends on the number of simultaneously computed output signals for a given input signal. To get meaningful performance figures despite these multiple degrees of freedom, two cases are considered here. In both examples, the total instruction rate R^{total} (6.3c) is shown for sampling frequencies $f_i = f_o = 1$. This choice is well suited for applications such as WFS that require continuously time-varying delays that change relatively slowly.

In Figure 6.16a, the total instruction rate R^{total} for a single output signal, that is $n = 1$, is shown for different implementation structures and design methods. It can be seen that the arithmetic complexity of the Oversampling+Lagrange structure designed with the overall optimization scheme proposed in Section 5.4 and the modified Farrow structure show similar behavior. The Oversampling+B-spline structure is comparably to these two structures up to medium values of A_s, but requires fewer instructions if the minimum stopband attenuation is high. In contrast, the Oversampling+OIB structure and the Oversampling+Farrow structure

(a) Storage requirement per input sample

(b) Memory bandwidth per output sample values/sample
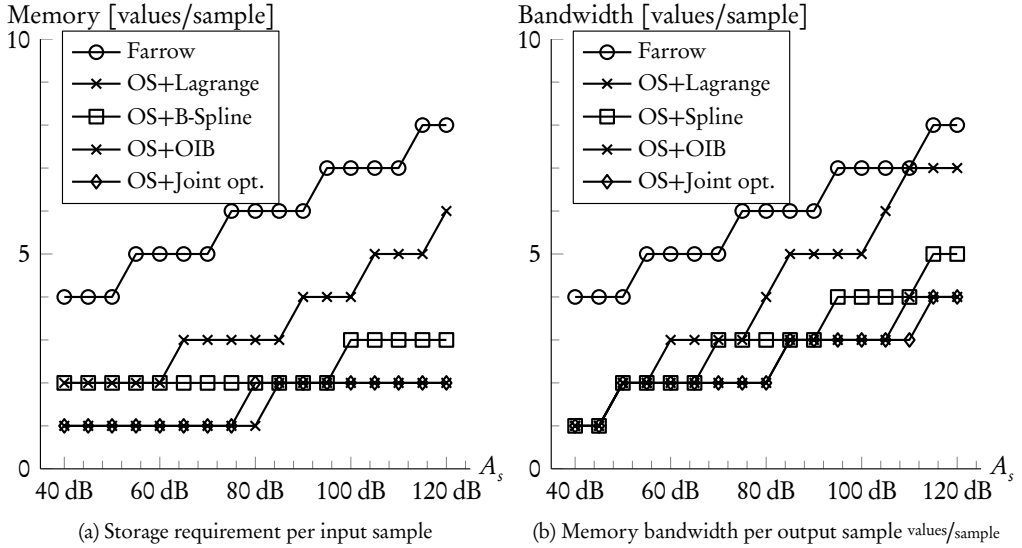
Figure 6.17: Minimum memory requirements for different ASRC implementation structures.

with iteratively optimized coefficients require a lower computational effort than the aforementioned structures for virtually all tested values of A_s. It is particularly notable that the efficient linear-complexity implementation structures for Lagrange interpolators do not result in a better performance of the Oversampling+Lagrange structure compared to other algorithms based on oversampling.

The same analysis is repeated in Figure 6.16b for a scenario where $n = 256$ output samples are computed for each input sample. This case resembles the application example of a WFS system with 256 loudspeaker channels introduced in Section 4.3.4. As argued there, the overall complexity is dominated by the effort to compute the output samples. Consequently, as the calculation of an output sample reduces to the evaluation of a polynomial in case of the modified Farrow structure, this structure shows a significantly lower instruction rate than the algorithms incorporating oversampling.

It is worth noting that in this use case, the complexity of the Oversampling+B-spline structure is higher than that of Oversampling+Lagrange. In contrast to the single-channel case, the reduced complexity of Lagrange interpolation results in a performance advantage if the overall effort is dominated by the computational costs for evaluating the resampling filter $H_{int}(j\omega)$. However, the efficiency of the Oversampling+Lagrange structure is inferior to structures using more sophisticated resampling functions, although the cost for evaluating these functions is comparable to that of a B-spline function of identical order.

6.4.2 Storage Requirements and Memory Bandwidth

For implementation on actual hardware, a performance evaluation purely based on the number of instructions is typically not sufficient. Performance metrics related to memory requirements

may prove decisive on many target architectures. The performance of an implementation structure with respect to such metrics might differ significantly from its arithmetic complexity.

As an example, a Farrow structure of order $M = 6$ is required to reach a stopband attenuation of about 95 dB for a cutoff frequency $\omega_c = 0.75\pi$ (see, for instance, Figure 6.1). Thus, the evaluation of one output sample requires only $2M = 12$ instructions, but $M + 1 = 7$ values have to be stored for each input sample, and the memory bandwidth required for the evaluation is also $M + 1 = 7$ values per output sample. In contrast, a structure based on oversampling with a jointly optimized resampling function with $L = 4$ and $N_{int} = 3$ attains the same quality with 26 instructions per output sample, but with a memory requirement of $L = 4$ values per input sample and a memory bandwidths of $N_{int} + 1 = 4$ accesses per output sample. For many applications and target platforms, the latter implementation structure is actually more appropriate due to limited storage capacity or the costs of memory accesses. In the same way, an evaluation based on the number of instructions favors structures with high oversampling ratios which are impractical for most applications due to excessive memory requirements.

For this reason, the trade-off analyses are repeated for the memory and bandwidth performance metrics introduced in Section 6.2. The graphs are shown in Figure 6.17. It is notable that these performance measures are independent of the output rate n, as they refer either to one input or to one output sample. As expected from the above argument, both the minimum memory requirements and the minimum bandwidths required by the Farrow structure are higher than those of the structures based on oversampling. For this class of structures, the memory requirements are determined by the oversampling ratio L, while the access bandwidth follows from the order N_{int} of the resampling function $H_{int}(j\omega)$. Thus, the performance gains of the designs proposed in Sections 5.5 and 5.6 result in significant reductions of the memory and bandwidth metrics compared to Oversampling+Lagrange structures.

6.4.3 Implementation Latency

The minimal implementation latency to reach a given stopband attenuation is displayed in Figure 6.18 for different ASRC implementation structures. It is observed that the dependency between the logarithmic measure A_s and the latency is approximately linear for all considered algorithms. Moreover, the required latencies are similar for all algorithms, while the modified Farrow structure typically gains a slight advantage. However, it is worth noting that the profoundly different nature of structures incorporating oversampling, namely the additional discrete-time prefilter and the operation on oversampled signals, does not have a more distinct effect on the overall latency.

Moreover, it is notable that the characteristic of the resampling function $H_{int}(j\omega)$ in structures incorporating oversampling has no significant impact on the minimum latency. Considering the results of Section 6.3.2.3, this implies that the increased maximum stopband attenuation offered by the more sophisticated resamplers do not have an effect if the objective is to minimize the implementation latency. On the other hand, this observation indicates that the advantageous performance characteristics of the optimized resampling filters proposed in this work can be used without increasing the latency.

In summary, these observations suggest that the latency requirements of an application have no decisive impact on the selection of an appropriate algorithm, because the latency required to achieve a desired performance level is very similar for all considered methods.
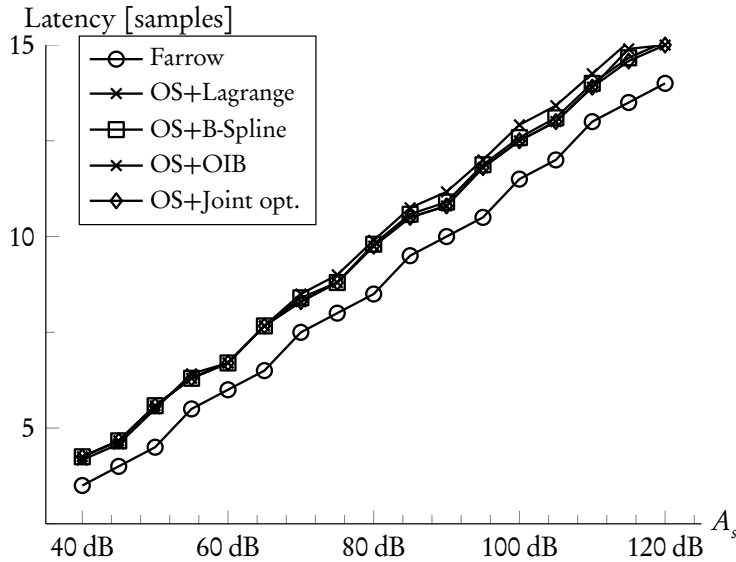
Figure 6.18: Minimum implementation latency for different ASRC implementation structures.

6.5 Conclusions

To assess the performance of the ASRC implementation structures considered in this thesis, design experiments for a large set of design parameters and objectives were performed in this chapter. In the first part, the effect of different parameter choices and design criteria on the achievable performance has been investigated. It turns out that the minimum stopband attenuation plays an outstanding role among the different quality measures. While other objectives or constraints are typically incorporated with relatively little impact on the computational effort, the required stopband attenuation has a direct and significant effect on the computational complexity.

In the second part, the computational resources required by the different implementation structure to reach a given design quality are investigated. With respect to the arithmetic complexity, Oversampling+OIB and Oversampling+Farrow structures are most efficient for single-channel applications. However, if a large number of output samples is computed for each input sample, the modified Farrow structure is significantly more efficient than algorithms based on oversampling.

However, this advantageous complexity of the modified Farrow structure is gained at the expense of rather large performance metrics for storage and memory bandwidth. It is shown that structures based on oversampling and sophisticated resampling functions offer vastly better performance with respect to these memory-related metrics.

Ultimately, the choice of a suitable algorithm depends on the properties of the hardware platform. The cost analyses performed in this chapter enable trade-offs between different performance aspects. In this way, they assist in the selection of an implementation structure that is appropriate for a given application and target architecture.

Chapter 7

Conclusions and Future Research

7.1 Summary

In this thesis, algorithms for arbitrary sample rate conversion (ASRC) and their application to wave field synthesis are investigated.

As argued in Chapter 2, the application of arbitrary, time-variant delays to discrete-time signals is one of the main signal processing tasks in practical real-time WFS reproduction systems. This operation is therefore of utmost importance for the computational cost of the complete system.

In Chapter 3, two models to describe this delay operation, variable fractional delay filtering (VFD) and arbitrary sample rate conversion (ASRC), are characterized. Corresponding design methods as well as implementation structures are reviewed. Particular emphasis is placed on a uniform notation, both for VFD and ASRC and for the representation of algorithms. This enables a comparison between different algorithms and brings out a number of similarities and equivalences as well as several general rules, such as the role of the basepoint set in utilizing symmetry relations. A second focus is placed on the computational complexity of different algorithms. For the case of Lagrange interpolation, numerous implementation structures are compared, and an efficient linear-complexity structure that performs favorably on modern superscalar, pipelined or explicitly parallel architectures is proposed. The VFD and ASRC models are compared, and ASRC turns out to be the more general approach to describe continuously time-variant delay operations.

Partly based on this insight, and in parts motivated by the specific requirements of WFS, it is reasoned in Chapter 4 why ASRC is the more appropriate approach to describe and implement the delay operations in a WFS signal processing system. To handle the complexity growth arising from the large number of simultaneous ASRC operations, a partitioning scheme for ASRC algorithms is proposed. This scheme takes the specific structure of a WFS signal processing system into account to reduce the overall computational complexity considerably. Conditions required by ASRC algorithms to comply to this partitioning scheme are stated. It is shown that several of the efficient algorithms described in Section 3.6 fit well into this structure.

Moving sound sources in WFS may result in both sample rate increases and decreases. Furthermore, the signal processing system is required to switch between these cases at arbitrary points in time and on a per-loudspeaker basis. While there is a general distinction between ASRC algorithms for increasing and decreasing the sample rate, the main problem for WFS is that algorithms for reducing the sampling rate cannot be integrated into the proposed partitioning scheme in an efficient way. However, it is demonstrated that this restriction does not pose an actual problem for the vast majority of WFS applications. For the remaining cases, a signal processing structure is outlined that enables arbitrary conversion ratios down to a configurable lower limit, necessitating only a moderate increase of the computational complexity.

Chapter 5 proposes several improvements to ASRC algorithms. Concise analytical descriptions for the continuous frequency responses of the modified Farrow structure and Lagrange interpolation are derived in Sections 5.2 and 5.3, respectively. In contrast to existing approaches, these expressions hold for arbitrary orders and enable insight into the characteristics of the respective structure. At the same time, they facilitate efficient and stable evaluation, which can be directly utilized in coefficient design and in the analysis of algorithms.

Coefficient design methods are described for the modified Farrow structure and for ASRC systems incorporating integer-ratio oversampling. Within this work, the main interest is on design methods that are optimal with respect to a given design specification, that is, an error norm and an optional set of constraints. A conforming design method is proposed for the widely-used Oversampling+Lagrange structure consisting of integer-ratio oversampling and Lagrange interpolation. Taking the frequency response of the overall system into account enables a reduction of the error norm as well as flexible, additional design specifications.

However, it turns out that the characteristics of Lagrange interpolation are not particularly well-suited for combination with oversampling. A set of alternative resampling functions to replace the Lagrange interpolator is proposed in Section 5.5. In conjunction with the proposed overall optimization scheme, these resampling filters enable gradual, significant performance improvements over Oversampling+Lagrange algorithms. In this way, the efficient implementation structures for Lagrange interpolation described in Section 3.6.3 appear to be the one of the few benefits of Lagrange interpolators when used in combination with oversampling.

Pushing this development to the extreme results in an ASRC structure consisting of oversampling and a freely parameterizable polynomial-based resampling function, termed Oversampling+Farrow. The design problem for this structure is non-convex. Thus, it can be solved neither globally optimal nor efficiently. Nonetheless, it is demonstrated that an iterative design method that optimizes the coefficients of both components alternately yields very good results if the initial solutions are chosen carefully. In this way, the fixed resampling functions proposed in Section 5.5 can be utilized advantageously for this more general class of ASRC filters. To assess the performance of these jointly optimized systems, a transformation is proposed that converts an Oversampling+Farrow algorithm into the so-called generalized Farrow structure, which is linear with respect to its coefficients. In this way, an upper bound for the performance of the Oversampling+Farrow structure is obtained. It is observed that for most practical design specifications, the performance of the Oversampling+Farrow structure comes very close to this theoretical upper limit.

In Chapter 6, the algorithms and implementation structures considered in this thesis are compared and analyzed with special emphasis on the arithmetic complexity and other performance metrics relevant to the appropriateness for a specific target platform. In addition to the dependency of the design objective on parameters such as the filter length or the order of the resampling functions, the effect of influencing variables such as the cutoff frequency of the input signal, fixed passband or stopband error bounds, or the use of nonuniform stopband specifications are considered. The results obtained are typically quite characteristic and can be used to state guidelines for choosing optimal design parameters for a given specification.

On the other hand, it becomes evident that the choice of an appropriate algorithm depends strongly on the characteristics of the target architecture. For applications such as WFS, where the number of processed output samples is large compared to the number of input samples, the modified Farrow structure is very efficient in terms of the instruction count. However, if criteria

such as storage requirements or memory bandwidth are considered, other algorithms such as integer-factor oversampling and jointly optimized resampling functions are often preferable. These results justify the trade-off analyses with respect to different performance metrics performed in this chapter. Moreover, they provide practical guidance on selecting an algorithm appropriate for a given application and target architecture.

In this way, the results obtained in this thesis are not limited to wave field synthesis, but might be applied to a broad range of applications of arbitrary sample rate conversion.

7.2 Directions for Future Research

While several improvements to ASRC algorithms have been proposed in this thesis, it also provides a number of worthwhile starting points for future research. Among these, the following appear particularly promising.

Analytical Performance Limits for Piecewise-Polynomial Resampling Functions

The performance analyses for the modified Farrow structures in Section 6.3.1 have shown several distinctive characteristics. While the achievable stopband performance mainly depends on the subfilter order M, it is relatively insensitive to design parameters such as the subfilter order, the cutoff frequency, or fixed passband error specifications. These observations suggest that a fixed limit for the stopband attenuation exists for each subfilter order. Therefore, it appears worthwhile to determine good estimates and to derive analytical error bounds. It is assumed that the analytical descriptions of the modified Farrow structure derived in Section 5.2, in particular the expression for the scaling function $\Phi(m, \omega)$ (5.23), provide a suitable basis for these investigations.

Globally Optimal Design of Oversampling+Farrow Structures

As stated in Section 5.6, the objective function for the coefficient design of Oversampling+Farrow structures is non-convex, thus prohibiting the use of global, efficient optimization methods. However, filter design techniques based on convex optimization have seen an impressive development in recent years (e.g. [WBV98, Dav10]). Numerous examples show that filter design problems which are non-convex in their original formulation can be often transformed into convex optimization problems. On the other hand, it has been shown that the class of Oversampling+Farrow structures is a subset of the class of generalized Farrow structures, which are linear with respect to their coefficients. For these reasons, it appears promising to investigate whether the design problem for the Oversampling+Farrow structure can be stated a convex, possibly nonlinear or constrained, optimization problem.

Efficient Implementation Structures for Fixed Resampling Functions for Use with Oversampling

Section 5.5 demonstrated that, compared to other piecewise-polynomial resampling kernels, Lagrange interpolators are not optimally suited for ASRC algorithms incorporating oversampling.

On the other hand, as described in Section 3.6.3, efficient linear-complexity implementation structures exist for Lagrange interpolation. In contrast, general piecewise-polynomial typically show an asymptotic complexity of $O(MN)$. Therefore, it is interesting to explore whether there are subsets of the class of piecewise polynomial function that can be implemented more efficiently than the general case while providing better stopband attenuation than Lagrange interpolation.

Design Criteria for Resampling Filters for Audio Applications

Within this thesis, the design specifications for ASRC algorithms were generally formed by objective criteria such as the norm of the frequency response error, optionally combined with time-domain constraints. These objectives are in accordance with most literature on SRC for audio applications, e.g. [LK81a, AK93]. In contrast, experiments have shown that the perceived audio quality does not necessarily coincide with these norms (see also [Ros93]).

Therefore, it appears worthwhile to investigate influencing factors for the perceived quality of ASRC algorithms in audio applications. The continuity or regularity of the continuous impulse response forms one starting point for this research. While strong discontinuities apparently result in audible artifacts, the time-domain continuity and regularity conditions stated in Section 5.2.4 appear too restrictive, as they result in a significant increase of the error norm. Pre-ringing artifacts of discrete-time filters (e.g. [Maa93]) are another example of time-domain effects. Moreover, the effect of quality criteria for VFD filters on the perceived quality of resampling filters forms a promising starting point.

Turning the obtained quality criteria into conditions and constraints that can be incorporated in a convex design specification would enable the purposeful design of ASRC structures specifically suited for audio applications.

List of Tables

List of Figures

Bibliography

[ACL93]   A. S. Alkhairy, K. G. Christian, and J. S. Lim. Design and characterization of optimal FIR filters with arbitrary phase. *IEEE Transactions on Signal Processing*, 41(2):559–572, February 1993.

[AGS10]   J. Ahrens, M. Geier, and S. Spors. Perceptual assessment of delay accuracy and loudspeaker misplacement in wave field synthesis. In *Proceedings of the 128th AES Convention*, London, UK, May 2010. AES.

[AK93]   R. Adams and T. Kwan. Theory and VLSI architectures for asynchronous sample-rate converters. *Journal of the Audio Engineering Society*, 41(7/8):539–555, July 1993.

[ARB04]   A. Apel, T. Röder, and S. Brix. Equalization of wave field synthesis systems. In *Proceedings of the 116th AES Convention*, Berlin, Germany, May 2004. AES.

[AS65]   M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions:*. Dover Publications, Inc., New York, NY, 1965.

[AS89]   J. L. Aravena and S. Soh. Architectures for polynomial evaluation. In *Proceedings of the Twenty-First Southeastern Symposium on System Theory*, pages 190–194. IEEE, March 1989.

[AS98]   J. Adams and J. Sullivan. Peak-constrained least-squares optimization. *IEEE Transactions on Signal Processing*, 46(2):306–321, February 1998.

[AS06]   W. A. Abu-Al-Saud and G. L. Stüber. Efficient sample rate conversion for software radio systems. *IEEE Transactions on Signal Processing*, 54(3):932–939, March 2006.

[AS07]   J. Ahrens and S. Spors. Implementation of directional sources in wave field synthesis. In *2007 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 66–69, New Paltz, NY, October 2007. IEEE.

[AS08]   J. Ahrens and S. Spors. Reproduction of moving virtual sound sources with special attention to the Doppler effect. In *124th AES Convention*, Amsterdam, The Netherlands, May 2008. AES.

[AUE92]   A. Aldroubi, M. Unser, and M. Eden. Cardinal spline filters: Stability and convergence to the ideal sinc interpolator. *Signal Processing*, 28(2):127–138, August 1992.

[AV00]   S. Akkarakaran and P. Vaidyanathan. Bifrequency and bispectrum maps: A new look at multirate systems with stochastic inputs. *IEEE Transactions on Signal Processing*, 48(3):723–736, March 2000.

[BBC76]   M. G. Bellanger, G. Bonnerot, and M. Coudreuse. Digital filtering by polyphase network: Application to sample-rate alteration and filter banks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-24(2):109–114, April 1976.

[BBS94]   C. S. Burrus, J. A. Bareto, and I. W. Selesnick. Iterative reweighted least squares design of FIR filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 42(11):2926–2936, November 1994.

[BBS04a]   K. Brandenburg, S. Brix, and T. Sporer. Wave field synthesis: From research to applications. In *Proceedings of the XII. European Signal Processing Conference*, pages 1369–1376, September 2004.

[BBS04b]   K. Brandenburg, S. Brix, and T. Sporer. Wave field synthesis: New possibilities for large-scale immersive sound reinforcement. In *Proceedings of the 18th International Congress on Acoustics*, volume I, pages 507–508, April 2004.

[BdVV93]   A. J. Berkhout, D. de Vries, and P. Vogel. Acoustic control by wave field synthesis. *Journal of the Acoustical Society of America*, 95(5):2764–2778, May 1993.

[Ber88]   A. Berkhout. A holographic approach to acoustic control. *Journal of the Audio Engineering Society*, 36(12):977–995, December 1988.

[BGR04]   D. Babic, A. Ghadam, and M. Renfors. Polynomial-based filters with odd number of polynomial segments for interpolation. *IEEE Signal Processing Letters*, 11(2):171–174, February 2004.

[Ble84]   N. Bleistein. *Mathematical Methods for Wave Phenomena*. Academic Press, Orlando, FL, USA, 1984.

[Ble96]   G. E. Blelloch. Programming parallel algorithms. *Communications of the ACM*, 39(3):85–97, March 1996.

[BLR03]   D. Babic, V. Lehtinen, and M. Renfors. Discrete-time modeling of polynomial-based interpolation filters in rational sampling rate conversion. In *Proceedings of the 2003 International Symposium on Circuits and Systems ISCAS '03*, volume 4, pages 321–324. IEEE, May 2003.

[Boo90]   C. d. Boor. Quasiinterpolation and approximation power of multivariate splines. In *Computation of Curves and Surfaces*. Kluwer, Dordrecht, Netherlands, 1990.

[Boo01a]   M. M. Boone. Acoustic rendering with wave field synthesis. In *ACM Siggraph and Eurographics Campfire*, Snowbird, UT, May 2001. ACM.

[Boo01b]   C. d. Boor. *A Practical Guide to Splines*. Applied Mathematical Sciences. Springer Verlag, New York, revised edition, 2001.

[Bra00]   R. N. Bracewell. *The Fourier Transform and its Applications*. McGraw-Hill, Boston, MA, 3rd edition, 2000.

[BSG92]   C. Burrus, A. Soewito, and R. Gopinath. Least squared error FIR filter design with transition bands. *IEEE Transactions on Signal Processing*, 40(6):1327 –1340, June 1992.

[BSMM06]   I. Bronstein, K. Semdjajew, G. Musiol, and H. Mühlig. *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Frankfurt am Main, Germany, sixth edition, 2006.

[BSSV89]   P. L. Butzer, K. Schmidt, E. Stark, and L. Vogt. Central factorial numbers; Their main properties and some applications. *Numerical Functional Analysis and Optimization*, 10:419–488, 1989.

[BT04]   J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.

[BTU99]   T. Blu, P. Thévenaz, and M. Unser. Generalized interpolation: Higher quality at no additional cost. In *Proceedings of the 1999 IEEE International Conference on Image Processing (ICIP'99)*, volume III, pages 667–671, Kobe, Japan, October 1999. IEEE.

[BTU01]   T. Blu, P. Thévenaz, and M. Unser. MOMS: Maximal-order interpolation of minimal support. *IEEE Transactions on Image Processing*, 10(7):1069–1080, July 2001.

[BTU03]   T. Blu, P. Thévenaz, and M. Unser. Complete parameterization of piecewise-polynomial interpolation kernels. *IEEE Transactions on Image Processing*, 12(11):1297–1309, November 2003.

[BU99]   T. Blu and M. Unser. Quantitative Fourier analysis of approximation techniques: Part I—interpolators and projectors. *IEEE Transactions on Signal Processing*, 47(10):2783 –2795, October 1999.

[Bur95]   C. Burrus. Multiband least squares FIR filter design. *IEEE Transactions on Signal Processing*, 43(2):412 –421, February 1995.

[BV04]   S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004.

[BVSR02]   D. Babic, J. Vesma, T. Saramäki, and M. Renfors. Implementation of the transposed Farrow structure. In *Proceedings of the 2002 IEEE Symposium on Circuits and Systems ISCAS 2002*, volume 4, pages 5–8. IEEE, May 2002.

[Can07]   c. g. Candan. An efficient filtering structure for Lagrange interpolation. *IEEE Signal Processing Letters*, 14(1):17–19, January 2007.

[CD10]   S. Chivapreecha and T.-B. Deng. Very low-complexity structure for Lagrange-type variable fractional-delay filter. In *2010 International Conference on Green Circuits and Systems ICGCS*, pages 137–141, Shanghai, China, June 2010. IEEE.

[CDPS91]   S. Cucchi, F. Desinan, G. Parladori, and G. Sicuranza. DSP implementation for arbitrary sampling rate conversion for high-quality sound applications. In *Proceeding of the 1991 International Conference on Acoustics, Speech and Signal Processing ICASSP-91*, Toronto, Canada, April 1991.

[CL00]   W. Cheney and W. Light. *A Course in Approximation Theory*. Brooks/ Cole, Pacific Grove, CA, 2000.

[CMT94]   G. Cain, N. Murphy, and A. Tarczynski. Evaluation of several variable FIR fractional-sample delay filters. In *Proceedings of the 1994 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP-94*, volume 3, pages 621–624, Adelaide, Australia, April 1994. IEEE.

[Coh95]   L. Cohen. *Time-Frequency Analysis*. Prentice Hall PTR, Englewood Cliffs, NJ, 1995.

[Cor07]   E. Corteel. Synthesis of directional sources using wave field synthesis, possibilities, and limitations. *EURASIP Journal on Advances in Signal Processing*, vol. 2007:18 pages, 2007. Article ID 90509.

[Cox72]   M. G. Cox. The numerical evaluation of B-splines. *Journal of the Institutes of Mathematics and its Applications*, 10(2):134–149, October 1972.

[Cox78]   M. G. Cox. The numerical evaluation of a spline from its B-spline representation. *Journal of the Institutes of Mathematics and its Applications*, 21:135–143, 1978.

[CR81]   R. E. Crochiere and L. R. Rabiner. Interpolation and decimation of digital signals — A tutorial review. *Proceedings of the IEEE*, 69(3):300–331, March 1981.

[CR83]   R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall Signal Processing Series. Prentice Hall, Inc., Englewood Cliffs, NJ, 1983.

[CS66]   H. Curry and I. Schoenberg. On Pólya frequency functions IV: The fundamental spline functions and their limits. *Journal Analyse Math.*, 17:71–107, 1966.

[CY03]   S. Chan and K. Yeung. On the application of variable digital filters (VDF) to the realization of software radio receivers. In *Proceedings of the 2003 International Symposium on Circuits and Systems ISCAS '03.*, volume 3, pages 562–565, Bangkok, Thailand, May 2003. IEEE.

[Dat97]   J. Dattorro. Effect design—part 2: Delay-line modulation and chorus. *Journal of the AES*, 45(10):764–788, October 1997.

[Dav10]   T. Davidson. Enriching the art of FIR filter design via convex optimization. *IEEE Signal Processing Magazine*, 27(3):89–101, May 2010.

[dB72]   C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, July 1972.

[dB76]   C. de Boor. Splines as linear combinations of B-splines: A survey. In G. G. Lorentz, C. K. Chui, and L. L. Schumaker, editors, *Approximation Theory*, volume II, pages 1–47. Academic Press, New York, NY, 1976.

[Den01]   T. B. Deng. Discretization-free design of variable fractional-delay FIR digital filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 48(6):637–644, June 2001.

[Den04]   T.-B. Deng. Symmetry-based low-complexity variable fractional-delay FIR filters. In *IEEE International Symposium on Communications and Information Technologies (ISCIT 2004)*, volume 1, pages 194–199, Sapporo, Japan, October 2004. IEEE.

[Den05]   T.-B. Deng. Accurate range for variable fractional-delay filtering. In *2005 Fifth International Conference on Information, Communications and Signal Processing*, pages 77–80. IEEE, December 2005.

[Den07a]   T.-B. Deng. Coefficient-symmetries for implementing arbitrary-order Lagrange-type variable fractional-delay filters. *IEEE Transactions on Signal Processing*, 55(8):4078–4090, August 2007.

[Den07b]   T.-B. Deng. Symmetric structures for odd-order maximally flat and weighted-least-squares variable fractional-delay filters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(12):2718–2732, December 2007.

[Den09]   T.-B. Deng. Robust structure transformation for causal Lagrange-type variable fractional-delay filters. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(8):1681–1688, August 2009.

[DH98]   C. Dick and F. Harris. FPGA interpolators using polynomial filters. In *The 8th International Conference on Signal Processing Applications and Technology*, Toronto, Canada, September 1998.

[DJR10]   J. Díaz-Carmona, G. Jovanovic-Dolecek, and A. Ramírez-Agundis. Frequency-based optimization design for fractional delay FIR filters with software-defined radio applications. *International Journal of Digital Multimedia Broadcasting*, 2010:1–6, 2010.

[DL06]   T.-B. Deng and Y. Lian. Weighted-least-squares design of variable fractional-delay FIR filters using coefficient symmetry. *IEEE Transactions on Signal Processing*, 54(8):3023–3038, August 2006.

[DLS02]   T. Davidson, Z.-Q. Luo, and J. Sturm. Linear matrix inequality formulation of spectral mask constraints with applications to FIR filter design. *IEEE Transactions on Signal Processing*, 50(11):2702–2715, November 2002.

[DM99]   A. G. Dempster and N. Murphy. Lagrange interpolator filters and binomial windows. *Signal Processing*, 76(1):81–91, July 1999.

[DM00]   A. Dempster and N. Murphy. The binomial window: Heuristics and metrics. *Signal Processing*, 80(12):2641–2645, December 2000.

[DN04]   T.-B. Deng and Y. Nagakawa. SVD-based design and new structures for variable fractional-delay digital filters. *IEEE Transactions on Signal Processing*, 52(9):2513–2527, September 2004.

[DRK93]   C. S. Dutta Roy and B. Kumar. Digital differentiators. In *Handbook of Statistics: Signal Processing and its Applications*, volume 10, chapter 6, pages 159–205. Elsevier Science Publishers, Amsterdam, 1993.

[DT96]   P. Depalle and S. Tassart. Fractional delay using Lagrange interpolators. In *Proceedings of the International Computer Music Conference ICMC'96*, pages 341–343, Clear Water Bay, Hong Kong, 1996. `http://recherche.ircam.fr/equipes/analyse-synthese/tassart/these/icmc96/paper.html`.

[DV77]   S. Dutta and M. Vidyasagar. New algorithms for constrained minimax programming. *Mathematical Programming*, 13(1):140–155, December 1977.

[dV09]   D. de Vries. *Wave Field Synthesis*. AES, 2009.

[EGH93]   L. Erup, F. M. Gardner, and R. A. Harris. Interpolation in digital modems — Part II: Implementation and performance. *IEEE Transactions on Communications*, 41(6):998–1006, June 1993.

[ESCA96]   K. Egiazarian, T. Saramäki, H. Chugurian, and J. Astola. Modified B-spline interpolators and filters: Synthesis and efficient implementation. In *Proceedings of the 1996 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP-96*, volume 3, pages 1743–1746. IEEE, May 1996.

[Eva00a]   G. Evangelista. Impulse response approximation for arbitrary sampling rate conversion. In *EUSIPCO 2000*, 2000.

[Eva00b]   G. Evangelista. *Zum Entwurf digitaler Systeme zur asynchronen Abtastratenumsetzung*. PhD thesis, Ruhr-Universität Bochum, October 2000. in german.

[Eva03]   G. Evangelista. Design of digital systems for arbitrary sampling rate conversion. *Signal processing*, 83(2):377–387, February 2003.

[Far88]   C. Farrow. A continuously variable delay element. In *1988 IEEE International Symposium on Circuits and Systems*, volume 3, pages 2641–2645, Espoo, Finland, June 1988. IEEE.

[FB09a]   A. Franck and K. Brandenburg. A closed-form description for the continuous frequency response of Lagrange interpolators. *IEEE Signal Processing Letters*, 16(7):612–615, July 2009.

[FB09b]   A. Franck and K. Brandenburg. An overall optimization method for arbitrary sample rate converters based on integer rate SRC and Lagrange interpolation. In *IEEE 2009 Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2009)*, New Paltz, NY, USA, October 2009. IEEE.

[FBR08]   A. Franck, K. Brandenburg, and U. Richter. Efficient delay interpolation for wave field synthesis. In *AES 125th Convention*, San Francisco, CA, October 2008. AES.

[FGKS07] A. Franck, A. Gräfe, T. Korn, and M. Strauß. Reproduction of moving sound sources by wave field synthesis: An analysis of artifacts. In *Proceedings of the AES 32nd International Conference*, Hillerød, Denmark, September 2007. AES.

[Fra08]   A. Franck. Efficient algorithms and structures for fractional delay filtering based on Lagrange interpolation. *Journal of the Audio Engineering Society*, 56(12):1036–1056, December 2008.

[Fra10]   A. Franck. A parallel linear-complexity implementation structure for Lagrange interpolation. In *2010 International Conference on Green Circuits and Systems ICGCS*, pages 158–163, Shanghai, China, June 2010.

[Gar93]   F. M. Gardner. Interpolation in digital modems — Part I: Fundamentals. *IEEE Transactions on Communications*, 41(3):501–507, March 1993.

[GB08]   M. C. Grant and S. P. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer, London, UK, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[GBLR04] A. S. H. Ghadam, D. Babic, V. Lehtinen, and M. Renfors. Implementation of Farrow structure based interpolators with subfilters of odd length. In *Proceedings of the 2004 International Symposium on Circuits and Systems, ISCAS '04.*, volume 3, pages 581–584, May 2004.

[GCB90]   A. Goshtasby, F. Cheng, and B. A. Barsky. B-spline curves and surfaces viewed as digital filters. *Computer vision, graphics, and image processing*, 52:264–275, November 1990.

[GKP06]   R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Boston, MA, 2nd edition, 2006.

[Gra04]   M. Grant. *Disciplined Convex Programming*. PhD thesis, Stanford University, CA, USA, 2004.

[Grä07]   A. Gräfe. Untersuchungen zu Frequenzverschiebungen bei der Wiedergabe bewegter Quellen in Wellenfeldsynthese. Diplomarbeit (diploma thesis), Technical University of Ilmenau, January 2007.

[Gre44]   T. N. Greville. The general theory of osculatory interpolation. *Transactions of the Actuarial Society of America*, 45:202–265, 1944.

[GS90]   W. B. Gearhart and H. S. Shultz. The function $\frac{\sin x}{x}$. *The College Mathematics Journal*, 21(2):90–99, March 1990.

[HA78]   H. S. Hou and H. C. Andrews. Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(6):508–517, December 1978.

[Har97]   F. Harris. Performance and design considerations of the Farrow filter when used for arbitrary resampling of time series. In *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems Computers*, volume 2, pages 1745–1749. IEEE, November 1997.

[Hec86]   P. S. Heckbert. Filtering by repeated integration. *SIGGRAPH Comput. Graph.*, 20(4):315–321, 1986.

[Hen79]   P. Henrici. Barycentric formulas for interpolating trigonometric polynomials and their conjugates. *Numerische Mathematik*, 33:225–234, 1979.

[Hen02]   T. Hentschel. *Sample Rate Conversion for Software Configurable Radio*. Artech House mobile communication series. Artech House Inc., 2002.

[Her71]   O. Herrmann. On the approximation problem in nonrecursive digital filter design. *IEEE Transactions on Circuit Theory*, 18(3):411–413, May 1971.

[Her92]   E. Hermanowicz. Explicit formulas for the weighting coefficients of maximally flat tunable FIR delayers. *Electronics Letters*, 28(20):1936–1937, September 1992.

[Her04]   E. Hermanowicz. On designing a wideband fractional delay filter using the Farrow approach. In *Proceedings of the 12th European Signal Processing Conference EUSIPCO-2004*, pages 961–964, Vienna, Austria, September 2004. EURASIP.

[HF00]   T. Hentschel and G. Fettweis. Continuous-time digital filtering for sample-rate conversion in reconfigurable radio terminals. In *Proceedings of the European Wireless*, pages 55–59, Dresden, Germany, September 2000.

[Hig04]   N. J. Higham. The numerical stability of barycentric Lagrange interpolation. *IMA Journal of Numerical Analysis*, 24(4):547–556, 2004.

[HJ05]   E. Hermanowicz and H. Johansson. On designing minimax adjustable wideband fractional delay FIR filters using two-rate approach. In *Proceedings of the 2005 European Conference on Circuit Theory and Design*, volume 3, pages 437–440. IEEE, August 2005.

[HRC73]   O. Herrmann, L. Rabiner, and D. Chan. Practical design rules for optimum finite impulse response low-pass digital filters. *Bell System Technical Journal*, 52(6):769–799, July–August 1973.

[Hul04]   E. Hulsebos. *Auralization using Wave Field Synthesis*. PhD thesis, Delft University of Technology, 2004.

[HV08]   A. Haghparast and V. Välimäki. A computationally efficient coefficient update technique for Lagrange fractional delay filters. In *Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2008*, pages 3737 – 3740, Las Vegas, NV, April 2008. IEEE.

[Jac02]   J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, NY, 3rd edition, 2002.

[Jan97]   G. Jansen. Focused wavefields and moving virtual sources by wavefield synthesis. Masters thesis, Delft University of Technology, Delft, Netherlands, 1997.

[Jer77]   A. J. Jerri. The Shannon sampling theorem — its various extensions and applications: A tutorial review. *Proceedings of the IEEE*, 65(11):1565–1596, November 1977.

[JL03]   H. Johansson and P. Löwenborg. On the design of adjustable fractional delay filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 50(4):164–169, April 2003.

[Kai74]   J. Kaiser. Nonrecursive digital filter design using the $i_0 -$ sinh window function. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 20–23, San Francisco, 1874. IEEE. reprinted in: Selected Papers in Digital Signal Processing, II, IEEE Press 1976.

[Key81]   R. G. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, December 1981.

[KG08]   F. M. Klingler and H. G. Göckler. Conversion between arbitrary sampling rates: An implementation cost tradeoff for the family of Farrow structures. In *Proceedings of the 5th Karlsruhe Workshop on Software Radios*, pages 1–8, Karlsruhe, Germany, 2008.

[KM99]   L. J. Karam and J. H. McClellan. Chebyshev digital FIR filter design. *Signal processing*, 79(1):17–36, July 1999.

[Kno96]   K. Knopp. *Theorie und Anwendung der unendlichen Reihen*. Springer Verlag Berlin, Germany, 6th edition, 1996.

[Knu98]   D. E. Knuth. *The Art of Computer Programming*, volume 2 - Seminumerical Algorithms. Addison Wesley Longman, third edition, 1998.

[KPPV07] M. Karjalainen, T. Paatero, J. Pakarinen, and V. Välimäki. Special digital filters for audio reproduction. In *Proceedings of the AES 32nd International Conference*, pages 91–108, Hillerød, Denmark, September 2007. AES.

[KW96]   P. J. Kootsookos and R. C. Williamson. FIR approximations of fractional sample delay systems. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(3):269–271, March 1996.

[Lag82]   R. Lagadec. Digital sampling frequency conversion. In *Procceedings of the 1st AES International Conference: Digital Audio*, Rye, NY, May 1982. AES.

[LBR04]   V. Lehtinen, D. Babic, and M. Renfors. Comparison of continuous- and discrete-time modeling of polynomial-based interpolation filters. In *Proceedings of the 6th Nordic Signal Processing Symposium NORSIG 2004.*, pages 49–52. IEEE, 2004.

[LD99]   W.-S. Lu and T.-B. Deng. An improved weighted least-squares design for fractional delay FIR filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(8):1035–1039, August 1999.

[Lev07]   A. Levitin. *Introduction to the Design & Analysis of Algorithms*. Addison Wesley, Boston, MA, 2nd edition, 2007.

[LK81a]   R. Lagadec and H. Kunz. A universal, digital sampling frequency converter for digital audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP '81*, volume 6, pages 595–598. IEEE, April 1981.

[LK81b]   R. Lagadec and H. O. Kunz. A new approach to digital sampling frequency conversion. In *68th AES Convention*, Hamburg, Germany, March 1981. AES.

[LLCY92]   Y.-C. Lim, J.-H. Lee, C. Chen, and R.-H. Yang. A weighted least squares algorithm for quasi-equiripple FIR and IIR digital filter design. *IEEE Transactions on Signal Processing*, 40(3):551–558, March 1992.

[LPW82]   R. Lagadec, D. Pelloni, and D. Weiss. A 2-channel, 16-bit digital sampling frequency converter for professional digital audio. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP '82.*, volume 7, pages 93 – 96, Paris, France, May 1982. IEEE.

[LR03]   V. Lehtinen and M. Renfors. Analysis of rational sample rate conversion using image response combining. In *Proceedings of the 2003 International Symposium on Circuits and Systems ISCAS '03*, volume 4, pages 325–328. IEEE, May 2003.

[LR09]   V. Lehtinen and M. Renfors. Structures for interpolation, decimation, and nonuniform sampling based on Newtonšs interpolation formula. In *Proceedings of the SAMPTA'09*, May 2009.

[LVKL96]   T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay: Tools for fractional delay filter design. *IEEE Signal Processing Magazine*, 13:30–60, January 1996.

[Maa93]   H. v. Maanen. Temporal decay: A useful tool for the characterization of resolution of audio systems. In *Proceedings of the 94th AES Convention*, Berlin, Germany, March 1993. AES.

[MDRK93]   S. Minocha, S. Dutta Roy, and B. Kumar. A note on the FIR approximation of a fractional sample delay. *International Journal of Circuit Theory and Applications*, 21(3):265–274, June 1993.

[Mei02]   E. Meijering. A chronology of interpolation: From ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, 90(3):319–342, March 2002.

[MKK94]   N. Murphy, A. Krukowski, and I. Kale. Implementation of a wideband fractional delay element. *Electronics Letters*, 30(20):1658–1659, September 1994.

[MKT97]   P. Murphy, A. Krukowski, and A. Tarczynski.   An efficient fractional sample delayer for digital beam steering. In *1997 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP-97*, volume 3, pages 2245–2248, April 1997.

[MP73]   J. H. McClellan and T. W. Parks. A unified approach to the design of optimum FIR linear-phase digital filters. *IEEE Transactions on Circuit Theory*, 20(6):697–701, November 1973.

[Mul97]   J.-M. Muller. *Elementary Functions: Algorithms and implementation*. Birkhäuser, Boston, MA, 1997.

[Nør24]   N. E. Nørlund. *Vorlesungen über Differenzenrechnung*. Julius Springer, Berlin, 1924.

[OSB99]   A. V. Oppenheim, R. W. Schafer, and J. R. Buck. *Discrete-Time Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[PB87]   T. W. Park and C. S. Burrus. *Digital Filter Design*. John Wiley & Sons, Inc., 1987.

[PM72a]   T. W. Parks and J. H. McClellan. Chebyshev approximation for nonrecursive digital filters with linear phase. *IEEE Transactions on Circuit Theory*, CT-19(2):189–194, March 1972.

[PM72b]   T. W. Parks and J. H. McClellan. A program for the design of linear phase finite impulse response filters. *IEEE Transactions on Audio and Electroacoustics*, AU-20(3):195–199, August 1972.

[Pot97]   A. W. Potchinkov. Design of optimal linear phase FIR filters by a semi-infinite programming technique. *Signal Processing*, 58(2):165–180, April 1997.

[Pow91]   M. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1991. Reprinted.

[PT98]   S.-C. Pei and C.-C. Tseng. A comb filter design using fractional-sample delay. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 45(5):649–653, May 1998.

[PT03]   S.-C. Pei and C.-C. Tseng. An efficient design of a variable fractional delay filter using a first-order differentiator. *IEEE Signal Processing Letters*, 10(10):307–310, October 2003.

[PTVF92]   W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[PVN+10]   J. Pekonen, V. Välimäki, J. Nam, J. O. Smith, and J. S. Abel. Variable fractional delay filters in bandlimited oscillator algorithms for music synthesis. In *Proceedings of the International Conference on Green Circuits and Systems*, 2010.

[Rab72]   L. R. Rabiner. Linear program design of finite impulse response (FIR) digital filters. *IEEE Transactions on Audio and Electroacoustics*, 20(4):280–288, October 1972.

[Ram84]   T. A. Ramstad. Digital methods for conversion between arbitrary sampling frequencies. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(3):577–591, June 1984.

[Ram98]   T. A. Ramstad. Fractional rate decimator and interpolator design. In *Proceedings of the Ninth European Signal Processing Conference EUSIPCO-98*, volume IV, pages 1949–1952, Rhodos, Greece, September 1998. EURASIP.

[RLF00]   K. Rajamani, Y.-S. Lai, and C. Farrow. An efficient algorithm for sample rate conversion from CD to DAT. *IEEE Signal Processing Letters*, 7(10):288–290, October 2000.

[Roc00]   D. Rocchesso. Fractionally addressed delay lines. *IEEE Transactions on Speech and Audio Processing*, 8(6):717–727, November 2000.

[Ros93]   D. Rossum. Constraint based audio interpolators. In *1993 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 161–164, New Paltz, NY, October 1993.

[Rus89]   J. J. Rushanan. On the Vandermonde matrix. *America Mathematical Monthly*, 96(10):921–924, December 1989.

[Rut76]   H. Rutishauser. *Vorlesungen über Numerische Mathematik*, volume 1. Birkhäuser, 1976. In german, english translation: *Lectures on Numerical Mathematics*, Walter Gautschi, ed., Birkhäuser, Boston, 1990.

[SA10]   S. Spors and J. Ahrens. Analysis and improvement of pre-equalization in 2.5-dimensional wave field synthesis. In *Proceedings of the 128th AES Convention*, London, UK, May 2010. AES.

[Sar93]   T. Saramäki. Finite impulse response filter design. In S. K. Mitra and J. F. Kaiser, editors, *Handbook for Digital Signal Processing*, chapter 4, pages 155–278. John Wiley & Sons, Inc., 1993.

[SAS04]   S. Samadi, M. O. Ahmad, and M. Swamy. Results on maximally flat fractional-delay systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(11):2271–2285, November 2004.

[Sch46a]   I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part A: On the problem of smoothing or graduation. A first class of analytic approximation formulae. *Quarterly of Applied Mathematics*, IV:45–99, 1946. Reprinted in: Carl de Boor (ed.): *I. J. Schoenberg – Selected Papers*, volume 2, Birkhäuser 1988.

[Sch46b]   I. J. Schoenberg. Contributions to the problem of approximation of equidistant data by analytic functions. Part B: On the problem of osculatory interpolation: A second class of approximation formulae. *Quarterly of Applied Mathematics*, IV:112–141, 1946. Reprinted in: Carl de Boor (ed.): *I. J. Schoenberg – Selected Papers*, volume 2, Birkhäuser 1988.

[Sch69] I. Schoenberg. Cardinal interpolation and spline functions. *Journal of Approxima-tion Theory*, 2(2):167–206, June 1969.

[Sch73] I. Schoenberg. *Cardinal Spline Interpolation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second printing 1993 edition, 1973.

[Sch74] I. Schoenberg. Cardinal interpolation and spline functions VII: The behavior of cardinal spline interpolants as their degree goes to infinity. *J. Analyse Math.*, XXVII:205–229, 1974.

[Sch97] H. R. Schwarz. *Numerische Mathematik*. Teubner, Stuttgart, Germany, 4th edition, 1997.

[Sch07] L. L. Schumaker. *Spline Functions: Basic Theory*. Cambridge University Press, Cambridge, UK, third edition, 2007.

[SF73]   G. Strang and G. Fix. A Fourier analysis of the finite element variational method. In *Constructive Aspects of Functional Analysis*, volume 1, pages 793–840. Edizione Cremonese, Rome, 1973.

[SG84]   J. O. Smith and P. Gossett. A flexible sampling-rate conversion method. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '84*, volume 9, pages 19.4.1–19.4.4, San Diego, CA, March 1984. IEEE.

[SH82]   H. W. Schüssler and H.J.Kolb. Variable digital filters:. *Archiv der Elektrischen Übertragung*, 36(6):229–237, 1982.

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, January 1949. Reprinted in: Proceedings of the IEEE, 86(2):447-457.

[SK97]   G. Stoyanov and M. Kawamata. Variable digital filters. *Journal of Signal Processing, Japan*, 1(4):275–290, July 1997.

[SLB96] I. Selesnick, M. Lang, and C. Burrus. Constrained least square design of FIR filters without specified transition bands. *IEEE Transactions on Signal Processing*, 44(8):1879–1892, August 1996.

[SM07]   M. Strauß and M. Munderloh. Influence of loudspeaker displacement on the reproduction quality of wave field synthesis systems. In *Proceedings of the 19th International Congress on Acoustics ICA 2007*, Madrid, Spain, September 2007.

[Smi07] J. O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, `http://www.w3k.org/books`, 2007. Online version `https://ccrma.stanford.edu/~jos/mdft/mdft.html` Accessed 2011-04-19.

[Smi10a] J. O. Smith. *Physical Audio Signal Processing*. W3K Publishing, 2010. online version `http://ccrma.stanford.edu/~jos/pasp/`, accessed 2011-02-24.

[Smi10b] J. O. Smith. *Spectral Audio Signal Processing, March 2010 Draft*. `http://ccrma.stanford.edu/~jos/sasp`, 2010. online book, accessed 2011-02-24.

[Smi11]   J. O. Smith. *Digital Audio Resampling Home Page*. `http://www-ccrma.stanford.edu/~{}jos/resample/`, February 2011. Accessed: 2011-02-04.

[Son00]   J.-J. Sonke. *Variable acoustics by wave field synthesis*. PhD thesis, Delft University of Technology, 2000. ISBN 90-9014138-3.

[SPB01]   T. Sporer, J. Plogsties, and S. Brix. CARROUSO - an european approach to 3D-audio. In *Proceedings of the 110th AES Convention*, Amsterdam, Netherlands, April 2001. Audio Engineering Society.

[Spo04]   T. Sporer. Wave field synthesis - generation and reproduction of natural sound environments. In *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, October 2004.

[SR73]   R. W. Schafer and L. R. Rabiner. A digital signal processing approach to interpolation. *Proceedings of the IEEE*, 61(6):692–702, June 1973.

[SR96]   T. Saramäki and T. Ritoniemi. An efficient approach for conversion between arbitrary sampling frequencies. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, volume 2, pages 258–288, Atlanta, GA, May 1996. IEEE.

[SRA08]   S. Spors, R. Rabenstein, and J. Ahrens. The theory of wave field synthesis revisited. In *Proceedings of the 124th AES Conference*, Munich, Germany, May 2008. AES.

[SSAB02]   J. O. Smith, S. Serafin, J. Abel, and D. Berners. Doppler simulation and the Leslie. In *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx-02)*, pages 13–20, Hamburg, Germany, September 2002.

[Sta97]   E. Start. *Direct Sound Enhancement by Wave Field Synthesis*. PhD thesis, Delft University of Technology, 1997.

[Str98]   H. Strauss. Implementing Doppler shifts for virtual auditory environments. In *104th AES Convention*, Amsterdam, The Netherlands, May 1998. AES.

[Str00]   H. Strauss. *Simulation instationärer Schallfelder für auditive virtuelle Umgebungen*. PhD thesis, Ruhr-Universität Bochum, Bochum, 2000. In german.

[SW70]   H. W. Schüssler and W. Winkelnkemper. Variable digital filters. *Archiv der Elektrischen Übertragung*, 24:524–525, November 1970.

[TCHR97]   A. Tarczyńsky, G. D. Cain, E. Hermanowicz, and M. Rojewski. WLS design of variable frequency response FIR filters. In *Proceedings of the 1997 IEEE International Symposium on Circuits and Systems*, pages 2224–2247. IEEE, June 1997.

[TD97]   S. Tassart and P. Depalle. Analytical approximations of fractional delays: Lagrange interpolators and allpass filters. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 1, pages 455–458, April 1997.

[Thi71]   J.-P. Thiran. Recursive digital filters with maximally flat group delay. *IEEE Transactions on Circuit Theory*, CT-18(6):659–664, November 1971.

[TKC94]   A. Tarczynski, W. Kozinski, and G. Cain. Sample rate conversion using fractional-sample delay. In *Proceedings of the 1994 International Conference on Acoustics, Speech and Signal Processing ICASSP-94*, volume 3, pages 285–288. IEEE, April 1994.

[Tse02]   C.-C. Tseng. Design of variable fractional delay FIR filter using differentiator bank. In *IEEE International Symposium on Circuits and Systems ISCAS 2002*, volume 4, pages 421–424. IEEE, May 2002.

[Tse04]   C.-C. Tseng. Design of variable fractional delay FIR filter using symmetry. In *Proceedings of the 2004 International Symposium on Circuits and Systems ISCAS '04*, volume 3, pages 477–480. IEEE, May 2004.

[Tse05]   C.-C. Tseng. Digital differentiator design using fractional delay filter and limit computation. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 52(10):2248–2259, October 2005.

[TYCT04]   K. M. Tsui, K. S. Yeung, S. C. Chan, and K. W. Tse. On the minimax design of passband linear-phase variable digital filters using semidefinite programming. *IEEE Signal Processing Letters*, 11(11):867–870, November 2004.

[UAE91]   M. Unser, A. Aldroubi, and M. Eden. Fast B-spline transforms for continuous image representation and interpolation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):277–285, March 1991.

[UAE93a]   M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part I—Theory. *IEEE Transactions on Signal Processing*, 41(2):821–833, February 1993.

[UAE93b]   M. Unser, A. Aldroubi, and M. Eden. B-spline signal processing: Part II—Efficient design and applications. *IEEE Transactions on Signal Processing*, 41(2):834–848, February 1993.

[Uns99]   M. Unser. Splines — A perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, November 1999.

[Uns00]   M. Unser. Sampling — 50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, April 2000.

[Vai84]   P. Vaidyanathan. On maximally-flat linear-phase FIR filters. *IEEE Transactions on Circuits and Systems*, 31(9):830–832, September 1984.

[Vai92]   P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice-Hall Inc, Englewood Cliffs, NJ, 1992.

[Väl95a]   V. Välimäki. *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters*. PhD thesis, Helsinki University of Technology, Espoo, Finland, 1995.

[Väl95b]   V. Välimäki. A new filter implementation strategy for Lagrange interpolation. In *1995 IEEE International Symposium on Circuits and System ISCAS '95*, volume 1, pages 361–364. IEEE, April 1995.

[Ver97]   E. Verheijen. *Sound Reproduction by Wave Field Synthesis*. PhD thesis, Delft
             University of Technology, 1997.

[Ves99]   J. Vesma. *Optimization and Applications of Polynomial-Based Interpolation Fil-
             ters*. PhD thesis, Tampere University of Technology, Department of Information
             Techhnology, Tampere, Finland, May 1999.

[Ves00]   J. Vesma. A frequency-domain approach to polynomial-based interpolation and
             the Farrow structure. *IEEE Transactions on Circuits and Systems II: Analog and
             Digital Signal Processing*, 47(3):206–209, March 2000.

[VH07]   V. Välimäki and A. Haghparast. Fractional delay filter design based on truncated
             Lagrange interpolation. *IEEE Signal Processing Letters*, 14(11):816 – 819, November
             2007.

[VHSR98]   J. Vesma, R. Hamila, T. Saramäki, and M. Renfors. Design of polynomial inter-
             polation filters based on Taylor series. In *Proceedings of the IX European Signal
             Processing Conference*, pages 283–286, Rhodes, Greece, September 1998.

[VLSR00]   J. Vesma, F. Lopez, T. Saramäki, and M. Renfors. The effects of quantizing
             the fractional interval in interpolation filters. In *Proceedings of the Nordic Signal
             Processing Symposium NORSIG2000*, pages 215–218, Kolmarden, Sweden, June
             2000.

[Vog93]   P. Vogel. *Application of Wave Field Synthesis in Room Acoustics*. PhD thesis, Delft
             University of Technology, 1993.

[VPEK06]   V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete-time modelling
             of musical instruments. *Reports on Progress in Physics*, 69(1):1–78, January 2006.

[VS96a]   J. Vesma and T. Saramäki. Interpolation filters with arbitrary frequency response
             for all-digital receivers. In *1996 IEEE International Symposium on Circuits and
             Systems ISCAS '96*, volume 2, pages 568–571. IEEE, May 1996.

[VS96b]   J. Vesma and T. Saramäki. Optimization and efficient implementation of fractional
             delay FIR filters. In *Proceedings of the Third IEEE International Conference on
             Electronics, Circuits, and Systems ICECS '96*, pages 546–549. IEEE, October 1996.

[VS97]   J. Vesma and T. Saramäki. Optimization and efficient implementation of FIR
             filters with adjustable fractional delay. In *Proceedings of the 1997 IEEE International
             Symposium on Circuits and Systems ISCAS '97*, volume 4, pages 2256–2259. IEEE,
             June 1997.

[VS00]   J. Vesma and T. Saramäki. Design and properties of polynomial-based fractional
             delay filters. In *Proceedings of the 2000 IEEE International Symposium on Circuits
             and Systems*, volume 1, pages 104–107, Genava, Switzerland, May 2000. IEEE.

[VS07]   J. Vesma and T. Saramäki. Polynomial-based interpolation filters—Part I: Filter
             synthesis. *Circuits Systems Signal Processing*, 26(2):115–146, April 2007.

[WBJ99]   D. K. Wise and R. Bristow-Johnson. Performance of low-order polynomial inter-
          polators in the presence of oversampled input. In *AES 107th Convention*, New
          York, NY, September 1999. Audio Engineering Society.

[WBV98]   S.-P. Wu, S. Boyd, and L. Vandenberghe. FIR filter design via spectral factorization
          and convex optimization. In B. Datta, editor, *Applied and Computational Control,
          Circuits and Systems*, volume 1, chapter 5, pages 1215–1248. Birkhäuser, Boston,
          MA, 1998.

[Whi15]   E. Whittaker. On the functions which are represented by expansions of the
          interpolation theory. *Proceedings of the Royal Society of Edinburgh*, 35:181–194,
          1915.

[Whi29]   J. Whittaker. On the cardinal function of interpolation theory. *Proceedings of the
          Edingburgh Mathematical Society*, 1(1):41–46, 1929.

[WR66]   E. T. Whittaker and G. Robinson. *The Theory of Observations*. Blackie and Sons
          Limited, London and Glasgow, fourth edition, 1966.

[Zad50]   L. A. Zadeh. Frequency analysis of variable networks. *Proceedings of the IRE*,
          38(3):291–299, March 1950.

[ZB94]   U. Zölzer and T. Boltze. Interpolation algorithms: Theory and application. In
          97th AES Convention, San Francisco, CA, November 1994.

[Zöl02]   U. Zölzer, editor. *DAFX*. Wiley & Sons, Chichester, UK, 2002.

Theses

I Wave Field Synthesis (WFS) is a computationally complex task. In realtime WFS reproduction systems, scaling and delaying of audio signals typically requires the majority of the computational effort.

II Synthesis of moving sound sources gives rise to several additional reproduction artifacts that do not occur in static WFS scenes. Artifacts due to time-variant delays form a major source of error. The required algorithms are best modeled as arbitrary sample rate conversion (ASRC) processes.

III In a WFS reproduction system, the number of time-variant delay operations increases linearly both with the number of sources and the number of loudspeakers. For this reason, the computational complexity of ASRC algorithms is of utmost importance. In most cases, the effort required for conventional high-quality algorithms is prohibitive.

IV The inherently parallel structure of a WFS signal processing system can be used to reduce the complexity of the delay operation significantly. For this purpose, the ASRC algorithms are partitioned into distinct stages that enable an efficient reuse of intermediate results. Several classes of efficient ASRC algorithms fit well into this partitioning scheme.

V Efficient, general-purpose algorithms for Lagrange interpolation can be implemented with linear complexity, opposed to $O(N^2)$ algorithms as predominantly used. While existing $O(N)$ algorithms are either not suited for general-purpose DSP applications or are highly sequential, a proposed implementation structure is well-suited for parallel, pipelined or superscalar architectures or for hardware implementation.

VI The performance of ASRC algorithms consisting of integer-ratio oversampling and Lagrange interpolation can be improved by applying an optimization method that takes the complete structure into account. This optimization technique enables optimal designs with respect to a selectable error norm and flexible, additional design constraints.

VII The proposed overall optimization scheme is applicable to other continuous-time resampling functions, yielding significant performance improvements compared to Lagrange interpolators. In particular, the use of resampling functions tailored to this system structure and the application of iterative joint optimization techniques yields a performance close to the theoretical upper bound for this structure for most design specifications.